

# SISTEM KENDALI JARAK JAUH BERBASIS MIKROKONTROLER AVR ATMEGA 8535 DI PABRIK KELAPA SAWIT

**Ahmad Mahfud**

Program Studi Pengolahan Hasil Perkebunan Kelapa Sawit

Politeknik Kelapa Sawit Citra Widya Edukasi – Bekasi

Email : [ahmad.mahfud@gmail.com](mailto:ahmad.mahfud@gmail.com)

## Abstrak

Perkembangan teknologi otomasi sudah mulai diterapkan di lingkungan industri perkelapasawitan secara menyeluruh. Keberadaan Pabrik Kelapa Sawit saat ini, merupakan kendala tersendiri dalam hal pemantauan terhadap operasionalnya. Proses kerja yang sangat bergantung pada performa setiap pekerja, menyebabkan tingkat variabilitas dalam pekerjaan sangat tinggi. Perlunya sistem monitoring terhadap kinerja pengolahan yang sedang berjalan secara *real time*, sangat mendukung tercapainya parameter dan standar-standar yang ditetapkan pada Pabrik Kelapa Sawit. Seorang *supervisor* sangat membutuhkan sebuah sistem monitoring yang mampu berfungsi sebagai sistem kendali untuk dapat mengakses kinerja pabrik dari jarak jauh, sehingga ia dapat memantau kinerja setiap mesin yang jumlahnya sangat banyak dan tempatnya berjauhan dengan area yang cukup luas, serta mampu dimonitor dan dikendalikan secara cepat dari satu lokasi yang sama. Pada penelitian ini akan diaplikasikan mikrokontroler AVR ATmega 8535 sebagai pengendali objek peralatan yang dapat dikontrol dari jarak jauh. Dengan menggunakan interface USB to *Serial Converter* serta perangkat *Laptop* dan *Gadget* yang terkoneksi dengan internet, *supervisor* akan dapat memonitor dan mengontrol operasional sebuah objek peralatan. Pemanfaatan AVR ATmega 8535 sangat memungkinkan dikembangkan untuk sistem kendali yang lebih kompleks. Visualisasi untuk sistem kendali yang lebih menarik dan mudah diterima oleh penggunaanya dengan memanfaatkan perangkat lunak Delphi 7.

## Kata Kunci

Sistem Kendali Jarak Jauh, AVR AT ATmega 8535, *USB to Serial*.

---

## Abstract

*Advances in automation technology have begun to be implemented in the palm oil industry in an integrated manner. The current existence of the Palm Oil Mill is an obstacle in terms of monitoring its operations. Work processes that are heavily dependent on the performance of each worker, cause the level of variability in work is very high. The need for a monitoring system on the performance of ongoing processing in real time, strongly supports the achievement of parameters and standards set at the Palm Oil Mill. A supervisor desperately needs a monitoring system capable of functioning as a control system in order to access mill performance remotely, so that he can monitor the performance of every machine that is very large and far from a large area, and can be monitored and controlled quickly from same location. In this research will be applied AVR AT ATmega 8535 mikrokontroler as controller of equipment object that can be controlled from long distance. By using the USB interface to Serial Converter as well as Laptop and Gadget devices connected to the internet, he will be able to monitor and control the operation of an equipment object. Utilization of AVR AT ATmega 8535 is possible to be developed for a more complex control system. Visualization for a more attractive and user-friendly control system can be made by using Delphi 7 software.*

## Keywords

*Remote Control System, AVR AT ATmega 8535, USB to Serial.*

## Pendahuluan



Proses pengolahan tandan buah kelapa sawit untuk diambil minyak mentahnya (*crude palm oil*) pada sebagian besar peralatan yang digunakan di dalam pabrik kelapa sawit, masih bersifat manual, di mana pengambilan keputusan dan pelaksanaannya semuanya tergantung pada peranan tenaga manusia. Hal ini berakibat pada tingkat kebutuhan sumber daya manusia yang membutuhkan jumlah yang relatif besar untuk beberapa peralatan. Belum lagi dengan kondisi mental dan fisik tenaga kerja yang sangat memungkinkan terjadinya fluktuasi, akan berakibat terjadinya variasi terhadap performa tenaga kerja dan hasil kerja.

Variabilitas pada hasil kerja akan sangat beresiko terhadap kenaikan produk yang dinilai tidak sesuai dengan kualifikasi produk yang diharapkan. Pada ujungnya akan berdampak pada kehilangan produksi dan kerugian yang lebih besar.

Untuk mengantisipasi kondisi tersebut, diperlukan sebuah sistem yang mampu dengan stabil menjalankan proses pengolahan secara otomatis dan mengurangi ketergantungan murni terhadap tenaga manusia. Sehingga walaupun terjadi variabilitas terhadap performa tenaga kerja, kualitas produk yang diharapkan tidak terlalu fluktuatif.

Di samping penerapan sistem otomasi yang diharapkan, juga terkait luasan area yang dapat dijangkau. Melihat banyaknya unit peralatan, beserta lokasi yang berpotensi bahaya tinggi serta area yang cukup luas sangat membutuhkan peran sebuah sistem yang mampu memonitor dan mengatur kinerja peralatan dari jarak yang relatif aman dari kondisi yang berbahaya.

Monitoring dan kontrol jarak jauh juga diperlukan untuk mengendalikan peralatan bila berada pada kondisi-kondisi darurat dimana posisi supervisor tidak berada dilokasi yang semestinya.

Dalam penelitian ini, dilakukan penelitian penggunaan mikrokontroler AVR ATmega 8535 sebagai *controller* yang berfungsi menerima sinyal masukan dari jarak jauh untuk kemudian menginstruksikan pada sebuah peralatan listrik (objek) untuk beroperasi. Sistem komunikasi jarak jauh menggunakan fasilitas sarana internet menggunakan *freeware Teamviewer* yang difungsikan untuk mengoperasikan PC dari jarak yang cukup jauh bahkan sangat jauh. Melalui *software Teamviewer*, seorang supervisor dapat mengakses program kontrolnya yang dibangun dari *platform Borland Delphi 7* dalam bentuk visual yang praktis, sederhana dan aplikatif.

## Landasan Teori

### **Mikrokontroler AVR ATmega 8535**

*Microcontroller AVR (Advance Virtual RISC)* pertama kali dikembangkan pada tahun 1996. Teknologi *Microcontroller AVR* menggunakan RISC (*Reduced Instruction Set Computer*). Dalam perkembangannya, AVR berkembang menjadi beberapa varian, di antaranya ATmega 8535.

---

Ahmad Mahfud

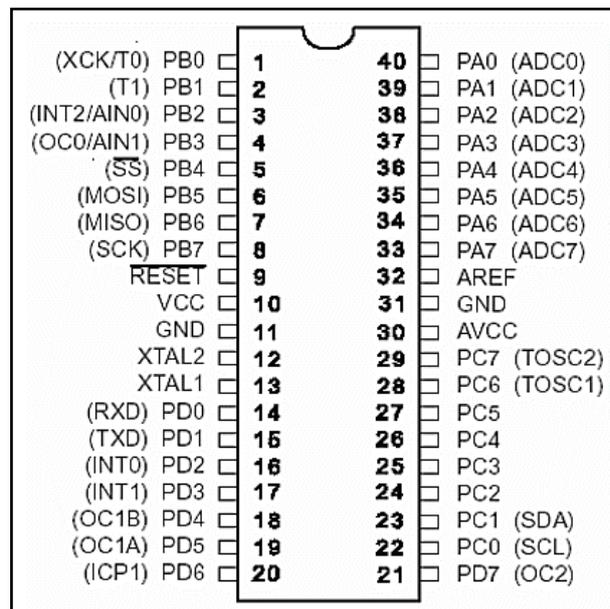
Sistem Kendali Jarak  
Jauh Berbasis  
Mikrokontroler AVR  
ATmega 8535 di Pabrik  
Kelapa Sawit

---

Karakteristik dari *microcontroller* AVR ATmega 8535 adalah sebagai berikut:

1. Frekuensi clock maksimum 16 MHz
2. I/O 32 buah.
3. ADC 10 bit, 8 input
4. Timer/Counter 3 buah
5. CPU 8 bit
6. SRAM 512 byte
7. Memori Flash 8 Kbyte
8. Port Komunikasi SPI
9. EEPROM 512 byte
10. USART 2,5 Mbps

Konfigurasi AVR ATmega 8535 dapat dilihat pada Gambar 1.



Gambar 1 Konfigurasi Pin ATmega 8535

## Borland Delphi 7

Delphi merupakan perangkat pengembangan aplikasi yang berjalan di sistem Windows. Perangkat ini merupakan kelanjutan dari Turbo Pascal yang merupakan produk Borland yang diluncurkan pertama kali pada tahun 1995.

Dalam pernyataannya, Borland mengatakan bahwa Delphi merupakan alat bantu yang dapat digunakan untuk *Rapid Application Development* (RAD). Dan kenyataannya hingga saat ini Delphi terbukti sebagai bahasa pemrograman yang paling baik. Selain sebagai aplikasi biasa, Delphi juga dapat digunakan untuk membuat aplikasi untuk internet. Sebagai contoh, kita dapat dengan mudah membuat sebuah aplikasi *ActiveForm* yang dapat berjalan pada Internet Explorer sebagaimana layaknya aplikasi pada umumnya.

Dalam hal pengolahan *database*, Delphi dapat mengakses *database* Paradox, MS-SQL, MySQL dan lain-lain. Sehingga kita dapat berpindah dari satu *database* ke *database* lain tanpa perlu mengubah aplikasi secara keseluruhan.

## Metode Penelitian

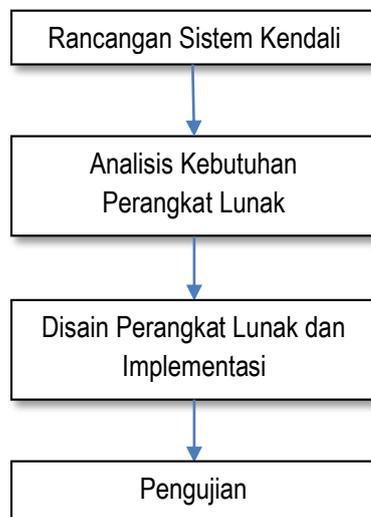
### Metode Pengumpulan Data

Untuk mengumpulkan data digunakan beberapa tahapan, yaitu:

1. Metode observasi, dengan mengadakan pengamatan langsung terhadap objek penelitian yaitu *Mikrokontroler AVR ATmega 8535*, perangkat lunak Borland Delphi 7 dan *Teamviewer*.
2. Metode Wawancara yang dilakukan dengan mengadakan tanya jawab kepada pihak yang terkait dengan objek yang diteliti.
3. Metode Kepustakaan, dengan menggali informasi dari buku, paper, dan sumber ilmiah lain, seperti situs internet ataupun artikel-artikel yang berkaitan dengan objek penelitian.

### Metode Pengembangan Perangkat Lunak

Diawali dengan rancangan sistem kendali, analisis kebutuhan perangkat lunak, disain perangkat lunak dan implementasi, dan dilanjutkan dengan pengujian. Urutan metode pengembangan tersebut dapat dilihat pada Gambar 2.



Gambar 2 Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak diutamakan untuk dapat membentuk komunikasi dengan serial port, baik untuk menerima sinyal input maupun memberikan sinyal output pada *port I/O*.

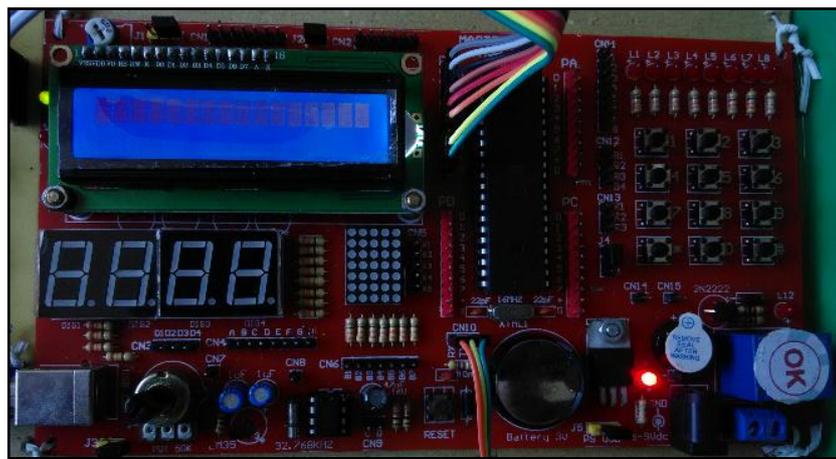
Program yang diterapkan pada Microkottroller AVR ATmega 8535 dipilih dengan menggunakan perangkat lunak CVAVR. Sedangkan perangkat

lunak yang digunakan untuk visualisasi di layar monitor menggunakan Delphi 7. Kedua perangkat lunak diarahkan untuk dapat berkomunikasi melalui *serial port*.

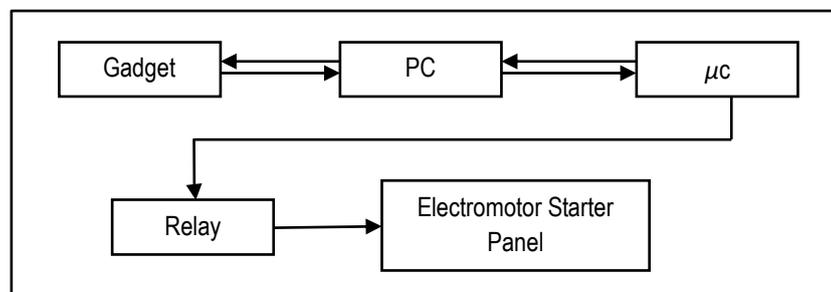
## Hasil Simulasi dan Pembahasan

### Rancangan Sistem Kendali

Hasil rancangan sistem kendali yang dibentuk menghubungkan antara *handphone* dengan PC/laptop dengan menggunakan fasilitas internet/WiFi. Berikutnya PC/Laptop akan memberikan sinyal kepada *Mikrokontroler* melalui *serial port*, untuk menghidupkan objek simulator.



Gambar 3 Rancangan Sistem Kendali



Gambar 4 Board Mikrokontroler AVR ATmega 8535

### Analisis Kebutuhan Perangkat Lunak

Sistem kendali dirancang agar dapat dimonitor dan dikendalikan dari jarak jauh, salah satu diantaranya dengan memanfaatkan fasilitas WiFi ataupun koneksi internet.

Namun demikian *controller* hanya akan bekerja lokal sebatas sistem kendali yang digerakkannya. Oleh karenanya, peneliti harus menggunakan/menggabungkan beberapa perangkat lunak agar dapat diterapkan secara serempak dan mendukung rancangan sistem kendali yang dibuat.

Untuk mengarahkan sistem kerja dari *controller* AVR ATmega 8535, digunakan perangkat lunak CV AVR. Sedangkan untuk pengambilan data dan visualisasinya dilayar monitor Laptop/PC digunakan Borland Delphi 7, dengan ditambahkan beberapa library agar Delphi 7 dapat mengakses I/O pada *serial port* yang ditentukan.

Untuk membaca tampilan dan memberikan masukan jarak jauh melalui laptop/pc/gadget, digunakan perangkat lunak penghubung *Teamviewer* yang memungkinkan pengguna mengakses seluruh fasilitas yang ada pada laptop/PC server.

## Desain Perangkat Lunak dan Implementasi

Desain perangkat lunak dan implementasinya meliputi:

### 1. Komunikasi *Mikrokontroler* dengan PC/Laptop

Untuk komunikasi ini menggunakan *serial port* pada *board mikrokontroler* yang selanjutnya akan dikoneksikan ke USB pada perangkat Laptop. Untuk koneksi ini membutuhkan instalasi USB to *serial* terlebih dahulu. Pemrograman *serial port* tersebut adalah sebagai berikut:

```
UCSRB=0b00011000;  
UCSRC=0b10000110;  
UBRRH=0;  
UBRRL=103;  
DDRB=0b11111111;
```

Pada UCSRB bit ke-4 bernilai 1 yang berarti RXEN = '1' atau dalam artian sistem penerimaan *serial* aktif.

Demikian pula untuk bit ke-3 bernilai 1 yang berarti TXEN = '1' atau sistem pengiriman data aktif.

### 2. Pembacaan dan Pengiriman Data *Serial Port Mikrokontroler*

Setelah proses inisialisasi *serial port* agar bisa terjalin komunikasi dua arah, berikutnya adalah pemrograman untuk pembacaan data dan pengiriman data dari dan ke laptop/komputer dengan program seperti terlihat pada Gambar 4.

Program pada Gambar 5 tersebut akan memerintahkan *controller* membaca data dari serial port dan memberikan output setelah ada data baru yang diterima.

Bila ada masukan data dari *port B* dengan nilai 00000000, maka *mikrokontroler* akan memberikan data output melalui perintah `printf` dengan data '0'. Demikian seterusnya.

```
while(1)
{
    b=getchar();
    switch(b)
    {
        case '0':PORTB=0b00000000; printf("0\r\n");break;
        case '1':PORTB=0b00000001; printf("1\r\n");break;
        case '2':PORTB=0b00000010; printf("2\r\n");break;
        case '3':PORTB=0b00000011; printf("3\r\n");break;
        case '4':PORTB=0b00000100; printf("4\r\n");break;
        case '5':PORTB=0b00000101; printf("5\r\n");break;
        case '6':PORTB=0b00000110; printf("6\r\n");break;
        case '7':PORTB=0b01000111; printf("7\r\n");break;
        case '8':PORTB=0b00001000; printf("8\r\n");break;
        case '9':PORTB=0b00001001; printf("9\r\n");break;
        case 'A':PORTB=0b00001010; printf("10\r\n");break;
        case 'B':PORTB=0b00001011; printf("11\r\n");break;
        case 'C':PORTB=0b00001100; printf("12\r\n");break;
        case 'D':PORTB=0b00001101; printf("13\r\n");break;
        case 'E':PORTB=0b00001110; printf("14\r\n");break;
        case 'F':PORTB=0b00001111; printf("15\r\n");break;
    }
}
```

Gambar 5 List Program Pembacaan dan Pengiriman *String* di *Mikrokontroler AVR*

### 3. Pemrograman Delphi 7 di PC/Laptop

Untuk pemrograman di PC/Laptop, terbagi menjadi beberapa tahapan, di antaranya:

#### a. Setting Alamat *Serial Port*

Untuk setting alamat *port* ini menggunakan program sebagai berikut:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    comport1.ShowSetupDialog;
end;
```

Gambar 6 List Program untuk Setting *Port IO* di Laptop

Pada prosedur diatas nampak bahwa bila tommol "1" ditekan, maka tampilan *setup port serial* akan muncul, sehingga memungkinkan bagi kita untuk memilih *port* yang sesuai.

#### b. Mengaktifkan komunikasi

Pengaktifan koneksi *serial port* diperlukan, agar bisa digunakan untuk jalur data masuk dan keluar.

```

begin
  comport1.Connected :=True;
  Button2.Caption :='Start';

```

Gambar 7 List Program untuk Mengaktifkan Port IO

List program di atas digunakan untuk mengaktifkan *comport1* dengan alamat yang sudah dimasukkan sebelumnya.

- c. Membaca data yang dikirimkan dari *mikrokontroler*  
 Untuk membaca informasi yang dikirimkan dari mikrokontroler, digunakan script program sebagai berikut:

```

var str:string;
begin
  comport1.ReadStr(str, count);
  memo1.Lines.Add(str);
end;

```

Gambar 8 List Program untuk Membaca dan Menampilkan Data yang Dikirim oleh Mikrokontroler

Data yang telah dibaca dengan *variable string*, selanjutnya dikirimkan ke kotak memo untuk ditambahkan satu baris *string* sebagai tulisan.

- d. Mengirim data ke *mikrokontroler*  
 Untuk mengirimkan data keluar, digunakan perintah *WriteStr*. Program selengkapnya sebagai berikut:

```

Label2.Caption:=x;
If x='0000' then comport1.WriteStr('0') else
If x='0001' then comport1.WriteStr('1') else
If x='0010' then comport1.WriteStr('2') else
If x='0011' then comport1.WriteStr('3') else
If x='0100' then comport1.WriteStr('4') else
If x='0101' then comport1.WriteStr('5') else
If x='0110' then comport1.WriteStr('6') else
If x='0111' then comport1.WriteStr('7') else
If x='1000' then comport1.WriteStr('8') else
If x='1001' then comport1.WriteStr('9') else
If x='1010' then comport1.WriteStr('A') else
If x='1011' then comport1.WriteStr('B') else
If x='1100' then comport1.WriteStr('C') else
If x='1101' then comport1.WriteStr('D') else
If x='1110' then comport1.WriteStr('E') else
  comport1.WriteStr('F');

```

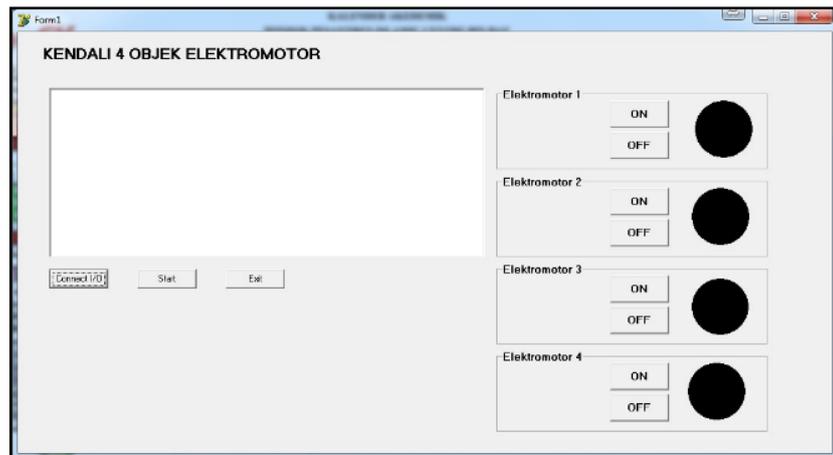
Gambar 9 Mengirimkan Data ke Mikrokontroler

## Pengujian

Tahap pengujian dilakukan setelah semua perangkat keras dan perangkat lunak terinstalasi dengan baik. Beberapa program dan *driver* juga sudah terinstal di dalam laptop. Pengujian yang dilakukan meliputi:

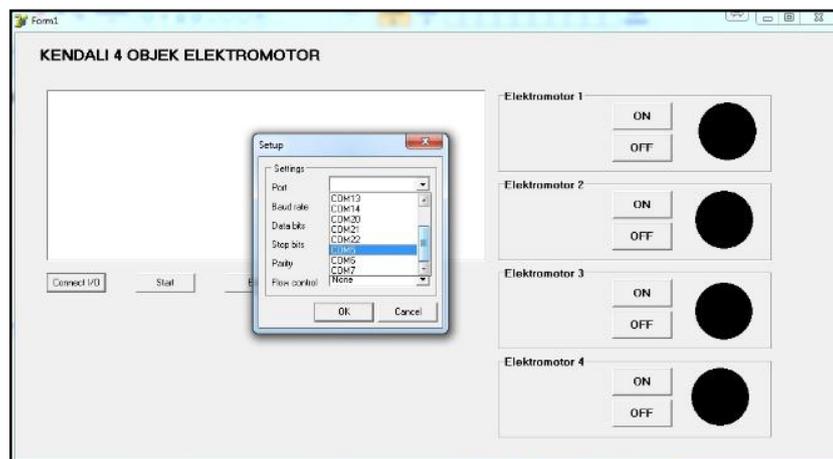
### 1. Pengujian Perangkat Lunak

Program yang dibangun dengan *platform* Delphi 7 diharapkan mampu memberikan tampilan sederhana, dan jelas. Serta mampu membaca dan memberikan output data ke *mikrokontroler* melalui *port* yang ditentukan.



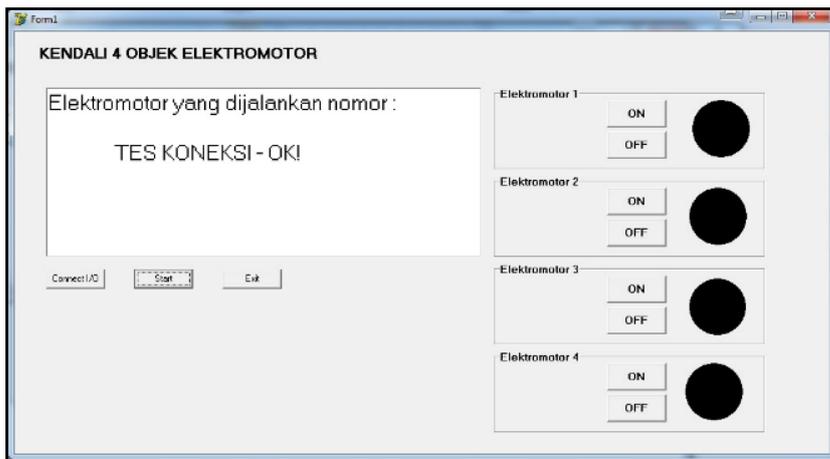
Gambar 10 Tampilan Program Kendali 4 Objek Elektromotor

### 2. Pengujian *Interface I/O* dari *serial port* dengan menggunakan perangkat *USB to Serial Converter*.



Gambar 11 Tampilan *Setting Port I/O*

Pada Gambar 11 di atas, dilakukan proses penentuan *port com5*, sebagaimana yang terdeteksi dari *device manager*. Penentuan ini sangat diperlukan untuk memberikan kepastian serah terima data dari dan ke luar Laptop.



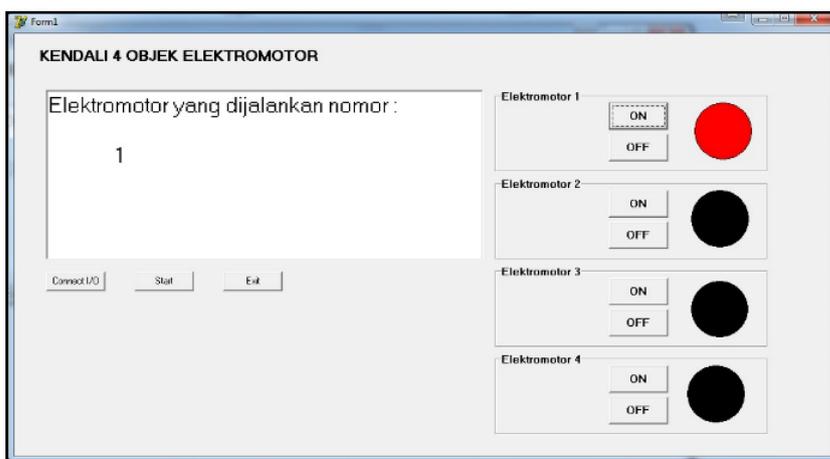
Gambar 12 Tampilan *Feedback* Hasil Koneksi dengan *Mikrokontroler*

Hasil pengujian koneksi antara laptop dengan *mikrokontroler* dengan menggunakan *USB to Serial Converter* telah berhasil. Data yang berupa tulisan (*String*) “TES KONEKSI – OK!” adalah *string* yang dikirimkan dari *mikrokontroler*.

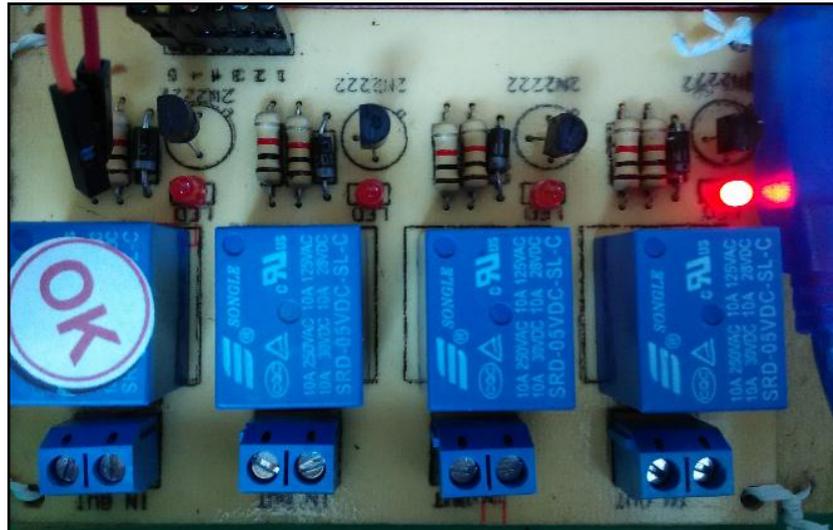
Dengan tertampalnya teks sebagaimana yang dikirimkan dari *mikrokontroler*, berarti komunikasi pengiriman data dari *microcontroller* dan pembacaan oleh Laptop telah berhasil.

3. Pengujian Pengendalian 4 elektromotor.

Pengujian pengendalian 4 elektromotor ini dilakukan dengan memberikan instruksi ON dan OFF melalui program yang sudah dibuat. Hasil kerja dimonitor langsung dari reaksi 4 buah *Relay* yang sudah dihubungkan dengan *mikrokontroler*. Respon *On/Off relay*, selanjutnya dikirimkan kembali ke *mikrokontroler* untuk ditampilkan ke dalam *display* (Memo) sebagai *On* Elektromotor yang diaktifkan.

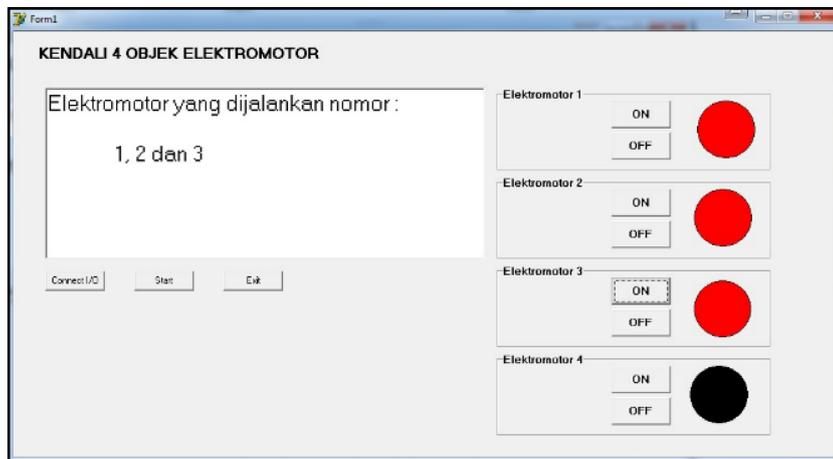


Gambar 13 Tampilan Saat Tombol *On* Elektromotor 1 Dijalankan

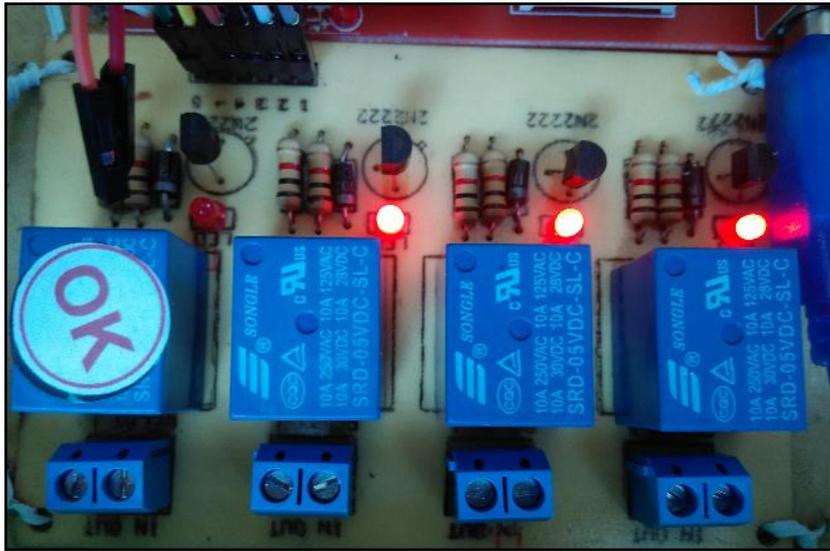


Gambar 14 Respon *Relay* 1 Saat Tombol *On* Elektromotor 1 Dijalankan

Gambar 13 dan 14 di atas menunjukkan hasil pengujian untuk respon pengaktifan *Relay* No. 1 yang diikuti dengan pengiriman kembali data bahwa *Relay* No. 1 sudah aktif dari *mikrokontroler* ke laptop. Sehingga laptop dapat memberikan informasi terkait *Relay* (*electromotor*) yang dijalankan.



Gambar 15 Tampilan saat tombol *On* Elektromotor 1, 2 dan 3 Dijalankan



Gambar 16 Respon Relay 1, 2 dan 3 Saat Tombol On Elektromotor 1, 2 dan 3 Dijalankan

## Kesimpulan

Berdasarkan hasil pembahasan dapat disimpulkan sebagai berikut:

1. Penggunaan *Mikrokontroler* AVR ATmega 8535 untuk sistem kendali jarak jauh, sangat memungkinkan dilakukan, dengan memanfaatkan perangkat pemrogram CV AVR.
2. Koneksi *interface* antara *Mikrokontroler* dengan Laptop dapat memanfaatkan *USB to serial converter*, namun membutuhkan pengkodean yang sesuai.
3. Proses pengiriman data dari *mikrokontroler* dengan meniadakan *bit paritas* dan *bit stop*, menyebabkan beberapa data terkadang terputus atau tidak lengkap.
4. Penggunaan Delphi 7 untuk visualisasi sistem kendali ini, sangat mudah dan memungkinkan untuk dikembangkan lebih jauh.

## Daftar Pustaka

- Joni, I.M., & Raharjo, B. (2006). *Pemrograman C dan Implementasinya*. Bandung: Penerbit Informatika.
- Green, D.C. (1996). *Komunikasi Data*. Yogyakarta: Andi.
- Bejo, A. (2008). *C & AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATmega 8535*. Semarang: Graha Ilmu.
- Sulasno, & Prayitno, T.A. (2008). *Teknik Sistem Kontrol*. Semarang: Graha Ilmu.
- Anonimous. (2015). *ATmega 8535 Data Sheet*.