

PERANCANGAN KONTROL DAN *MONITORING* KECEPATAN MOTOR DC MELALUI JARINGAN INTRANET

Azwardi

Jurusan Teknik Komputer Politeknik Negeri Sriwijaya, Palembang,
email:sylvanums@gmail.com

Abstrak—Perkembangan aplikasi yang menggunakan mikrokontroler semakin berkembang pesat, diantaranya aplikasi mikrokontroler sebagai alat kontrol dan monitoring suatu proses dalam industri, untuk mengontrol dan me-monitoring dalam jarak jauh dapat digunakan berbagai media komunikasi diantaranya adalah jaringan intranet/ internet. Jaringan intranet/ internet mempunyai keunggulan dapat dijadikan sebagai media transmisi berbagai data dari berbagai peralatan dan mempunyai jangkauan yang luas. Pada makalah ini dibahas tentang perancangan sistem kontrol dan monitoring kecepatan motor DC melalui jaringan intranet. Sistem dibuat menggunakan mikrokontroler jenis AVR, agar mikrokontroler dapat berfungsi sebagai server yang mempunyai alamat IP sendiri yang dapat berkomunikasi secara langsung melalui jaringan intranet dengan suatu komputer master kontrol maka diperlukan suatu IC kontroler. IC W3100A digunakan sebagai kontroler ini yang dapat menerapkan protokol TCP/IP didalamnya. Pada komputer master kontrol dibuat program client yang akan mengirim data setting dan menerima data monitoring kecepatan motor dari mikrokontroler. Untuk mempercepat respon perubahan kecepatan motor terhadap nilai setting kecepatan dan meminimalkan error kecepatan motor maka sistem yang dibuat menerapkan sistem kontrol tertutup dengan PID sebagai kontrolernya.

Kata Kunci: monitoring, kontrol, intranet

1. PENDAHULUAN

Dengan semakin pesatnya perkembangan internet di berbagai wilayah maka kita dapat berkomunikasi dengan jarak yang sangat jauh dan melakukan proses bisnis diantaranya *tebanking*, mengirim *email*, menerima *email*, serta *shopping* dari berbagai tempat di seluruh dunia yang terdapat jaringan internetnya.

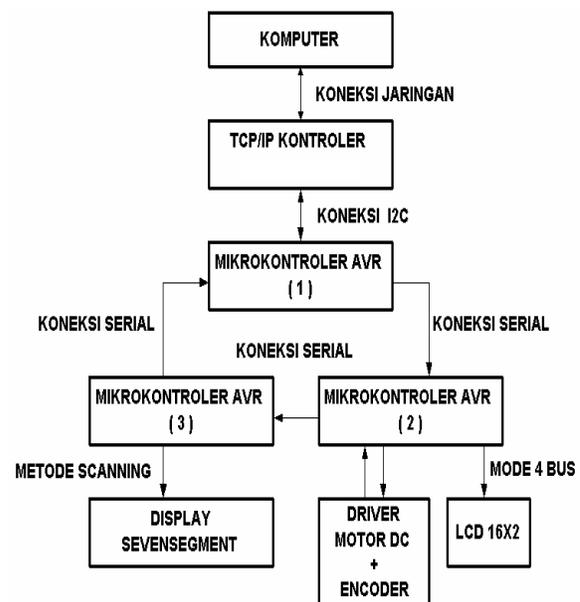
Pada suatu Industri yang terdapat jaringan intranet/internetnya, Industri tersebut dapat menggunakan jaringan tidak hanya sebagai media komunikasi antara komputer dengan komputer tetapi dapat juga digunakan untuk media komunikasi antara mikrokontroler sebagai alat kontrol dan komputer sebagai *master* kontrol.

Dengan digunakannya jaringan intranet/ internet dapat diperoleh berbagai keuntungan diantaranya penghematan media jaringan yaitu kabel yang lebih sedikit, dapat diakses dari berbagai tempat yang ada jaringan internetnya serta lebih menghemat tempat karena ukuran mikrokontroler yang sangat kecil dibandingkan dengan komputer.

2. PERANCANGAN SISTEM

Sistem ini memiliki enam komponen utama yaitu komputer master kontrol, kontroler TCP/IP, 3 buah mikrokontroler jenis AVR, penampil LCD, penampil

seven segment, dan motor DC. Diagram blok komponen utama ditunjukkan pada Gambar 1.



Gambar 1. Diagram blok sistem

Setiap bagian diagram blok tersebut dapat dijelaskan sebagai berikut :

- Pada komputer master kontrol dibuat suatu program *client* yang memberikan nilai *setting* kecepatan motor serta *me-monitor* besarnya kecepatan motor yang dihasilkan dan menyimpannya dalam *database*.
- TCP/IP kontroler menerima *setting register* dari mikrokontroler AVR (1) sehingga dapat mengambil data yang ada di jaringan yang dialamatkan ke alamat IP alat yang telah ditentukan oleh mikrokontroler AVR (1), komunikasi antara mikrokontroler AVR (1) dan TCP/IP kontroler menggunakan komunikasi I2C.
- Data dari jaringan diproses oleh mikrokontroler AVR (1) dan data *setting* kecepatan dikirim ke mikrokontroler AVR (2) menggunakan komunikasi serial. Oleh mikrokontroler AVR (2) data *setting* kecepatan ditampilkan di LCD dan kemudian diproses melalui PID digital kemudian dikonversi menjadi sinyal PWM (*Pulse Width Modulation*), sinyal pwm ini digunakan sebagai input dari rangkaian *driver* motor
- Saat motor berputar rangkaian *Encoder* yang ada di motor menghasilkan pulsa yang dihitung oleh mikrokontroler AVR (2) setiap detik, hasil perhitungan ini ditampilkan ke LCD dan dikirim ke mikrokontroler AVR (3). Data perhitungan kecepatan motor yang telah diterima oleh mikrokontroler AVR (3) ditampilkan ke 4 buah penampil *sevensegment* dengan metode *scanning* dan dikirim ke mikrokontroler AVR (1) dengan komunikasi serial yang selanjutnya akan dikirim ke komputer master kontrol melalui jaringan.

2.1 Perancangan perangkat lunak

- Perangkat lunak yang dibuat ada 4 buah yaitu:
- Untuk mikrokontroler AVR (1) yang digunakan sebagai pengontrol dan pengambil data dari IC TCP/IP kontroler, flowchart perangkat lunak yang dibuat untuk mikrokontroler AVR (1) ditunjukkan Gambar 2.

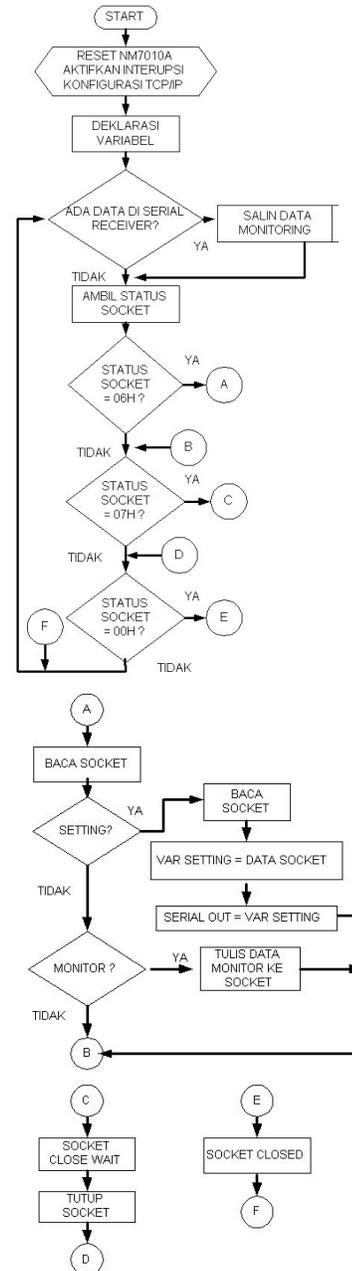
Dalam sistem ini komunikasi yang diterapkan adalah jenis komunikasi *Passive Open* yaitu mengkonfigurasi alat yang dibuat sebagai *server* sehingga alat selalu memeriksa adanya sinyal SYN yang ditujukan ke alat yang berasal dari jaringan. Langkah-langkah yang dilakukan dalam mode ini adalah :

- **CLOSED STATE** : inisialisasi *channel* dengan mengeksekusi perintah *sys_init* atau perintah *Close*
- **INIT STATE** : mengatur jumlah *port* yang digunakan dalam *channel* dengan mengeksekusi perintah *sock_init*
- **LISTEN STATE** : selalu menunggu koneksi dari *peer* (menunggu sinyal SYN), setelah sinyal SYN pada port yang ditentukan diterima maka paket SYN, ACK ditranmisikan dan berganti ke SYNRCVD STATE

Dalam SYNRCVDSTATE setelah paket SYN, ACK dikirim maka IC W3100A menunggu balasan sinyal ACK dari *peer*, jika balasan tersebut diterima maka berganti ke ESTABLISED STATE.

Gambar 3 menunjukkan ilustrasi dari mode *server*. Setelah berada dalam ESTABLISED STATE maka program melakukan *looping* terus-menerus dan memeriksa keadaan *socket*, ada 3 macam keadaan *socket* yaitu :

1. **SOCK_ESTABLISHED (06H)** yaitu keadaan koneksi telah terjadi tetapi dalam mode pasif

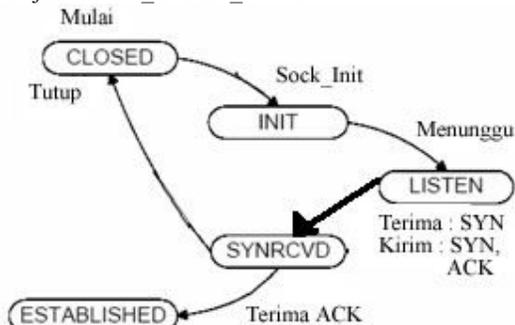


Gambar 2 Flowchart AVR (1)

2. *SOCK_CLOSE_WAIT* (07H) yaitu koneksi telah terputus

3. *SOCK_CLOSED* (00H) yaitu koneksi diputus
 Jika dalam keadaan *SOCK_ESTABLISHED* maka dilakukan pembacaan *socket*, data *setting* kecepatan yang diterima disalin ke variabel untuk dikirim ke mikrokontroler AVR(2) melalui komunikasi serial, sedangkan jika program *client* yang ada di komputer meminta data hasil *monitoring* maka dilakukan penulisan data hasil *monitoring* ke *socket* untuk selanjutnya dikirim ke *client* melalui jaringan.

Jika koneksi yang terjadi telah terputus (*SOCK_CLOSE_WAIT*) maka *socket* ditutup, dan keadaan *SOCK_CLOSED* adalah keadaan *socket* memang sengaja ditutup oleh mikrokontroler AVR(1)/ *server* ataupun oleh program yang ada di komputer/ *client*. Pemeriksaan keadaan *socket* dilakukan terus-menerus karena protocol TCP/IP adalah protocol yang *Connection Oriented* yaitu koneksi yang memeriksa data yang telah kirim ke *client* dan menunggu balasan dari *client* bahwa data yang dikirim telah diterima, jika dalam waktu yang ditentukan belum ada balasan maka data dicoba untuk dikirim ulang, belum adanya balasan dari *client* menjadikan keadaan *socket* menjadi *SOCK_CLOSE_WAIT*.



Gambar 3 Ilustrasi mode Server

b. Untuk mikrokontroler AVR (2) digunakan sebagai pembangkit PWM dengan kontroler PID dan penghitung pulsa dari *encoder* serta penampil LCD, flowchart perangkat lunak untuk mikrokontroler AVR(2) ditunjukkan gambar 4.

Penghitungan pulsa dilakukan selama satu detik agar menghasilkan frekuensi dalam satuan *Hertz*, Dalam alat ini jumlah lubang dalam *rotary encoder* ada 30 buah sehingga untuk mengkonversi satuan kecepatan menjadi dalam RPM (Round Per Minute)/ jumlah putaran tiap menit maka dilakukan penghitungan sebagai berikut :

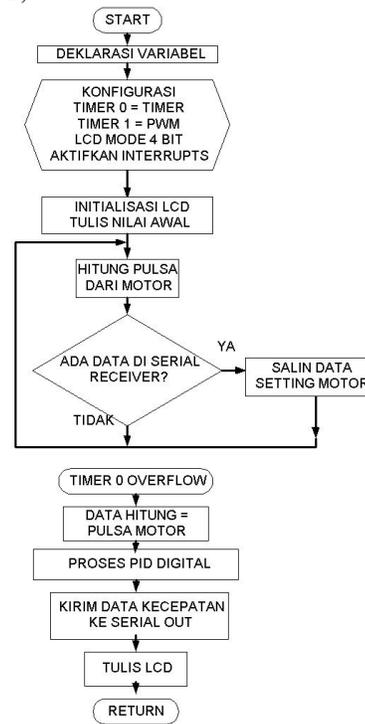
$$RPM = (\text{hasil penghitungan pulsa} / 30) \times 60$$

$$= \text{hasil penghitungan pulsa} \times 2$$

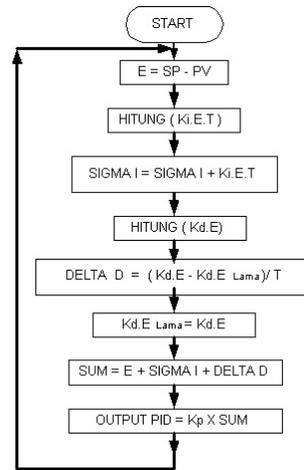
Proses PID secara digital seperti ditunjukkan oleh gambar 5.

Penjelasan *Flowchart* PID Digital adalah sebagai berikut:

- Error (E) adalah selisih antara *Set Point* (SP) dengan *Proses Variabel* (PV)
- Di hitung nilai $K_i.E.T$ dengan K_i adalah Konstanta Integral yang dalam tugas akhir ini dicari dengan menggunakan metode ekperimental dan T adalah periode *sampling* yang dalam tugas akhir ini adalah 1 detik.
- Dihitung nilai integral $\sum_i = \sum_i + K_i.E.T$
- Dihitung nilai $K_d.E$ dengan K_d adalah Konstanta Diferensial yang dalam tugas akhir ini juga dicari dengan menggunakan metode ekperimental.
- Dihitung nilai Diferensial $\Delta D = (K_d.E - K_d.E \text{ lama}) / T$,

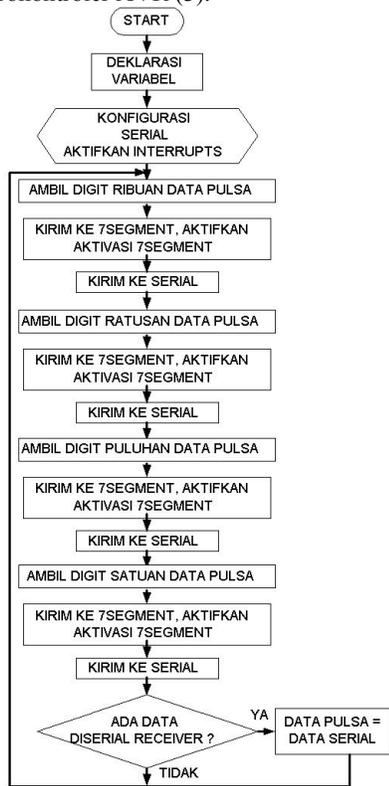


Gambar 4 Flowchart AVR (2)



Gambar 5 Flowchart PID Digital

- Kd.E lama adalah nilai Kd.E proses penghitungan sebelumnya sehingga agar bisa digunakan untuk proses selanjutnya maka nilai Kd.E sekarang disimpan dengan perintah $Kd.E\ lama = Kd.E$
 - Penjumlahan total adalah $SUM = E + \sum i + \Delta D$
 - *Output* dari PID adalah hasil perkalian K_p dengan SUM , dengan K_p adalah Konstanta Proporsional. K_p dalam tugas akhir ini juga dicari dengan menggunakan metode ekperimental
- c. Untuk mikrokontroler AVR (3) yang digunakan sebagai penampil *seven segment* dengan metode *scanning*. Gambar 6 menunjukkan flowchart dari mikrokontroler AVR (3).



Gambar 6 Flowchart AVR (3)

- d. Sedangkan perangkat lunak yang ada di komputer master kontrol sebagai *client* dari alat yang dibuat dirancang menggunakan bahasa pemrograman delphi 7. flowchart perangkat lunak pada komputer master kontrol ditunjukkan oleh gambar 7.

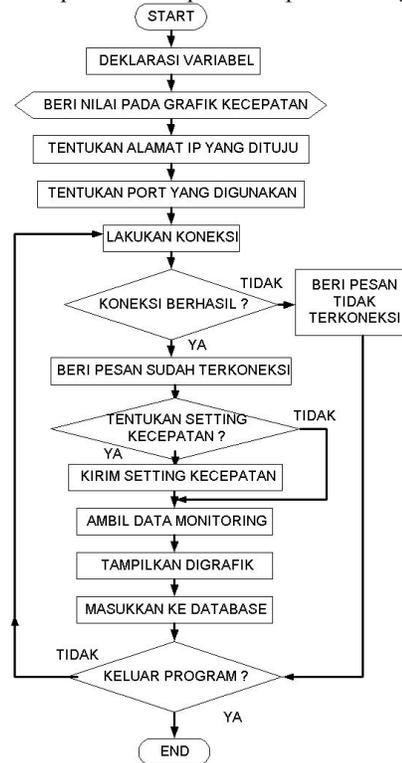
3 Pengujian Alat

Bagian-bagian sistem yang diuji antara lain : transmisi data melalui jaringan antara alat yang dibuat dengan komputer *master* kontrol, dan pengujian respon motor terhadap setting kecepatan.

- Pengujian perbandingan data yang diterima dengan data yang dikirim dalam jaringan dilakukan dengan konfigurasi *peer to peer*, melalui *switch*, dan melalui *hotspot*. Dari ketiga

konfigurasi tersebut didapatkan hasil yang sama seperti ditunjukkan tabel 1 dengan data dikirim dari alat dan tabel 2 dengan data dikirim dari komputer *master* kontrol.

- Pengujian respon kecepatan motor terhadap *setting* kecepatan dilakukan untuk mengetahui respon kecepatan motor serta karakteristik kontroler PID yang dibuat dalam menyamakan kecepatan motor dengan kecepatan *setting*. Dari eksperimen didapat bahwa untuk kecepatan 2500 rpm, K_p yang sesuai sebesar 0.5, $K_i = 0.02$ dan $K_d = 0.01$, Pengujian dilakukan dengan melihat respon kecepatan motor pada kecepatan *setting*



Gambar 7 Flowchart Program Client

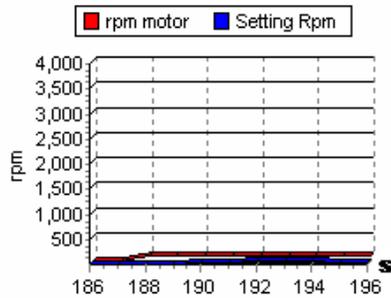
Tabel 1 Hasil transmisi data dari alat ke Client

Pengiriman Ke	Data dikirim dari alat (rpm)	Data diterima di komputer client (rpm)
1	0	0
2	1	1
3	10	10
4	50	50
5	100	100
6	500	500
7	1000	1000
8	1525	1525
9	2010	2010
10	3500	3500

Tabel 2 Hasil transmisi data dari Client ke Alat

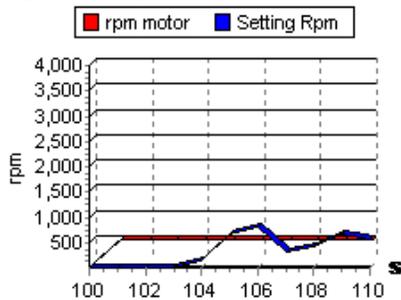
Pengiriman Ke	Data dikirim dari Client (rpm)	Data diterima di alat (rpm)
1	0	0
2	1	1
3	10	10
4	50	50
5	100	100
6	500	500
7	1000	1000
8	1500	1500
9	2000	2000
10	3500	3500

100 rpm, 500 rpm, 1000 rpm, 1500 rpm, 2000 rpm, 2500 rpm, 3000 rpm, dan 3500 rpm. Gambar respon dari kecepatan motor pada setting kecepatan 100 rpm ditunjukkan gambar 8.



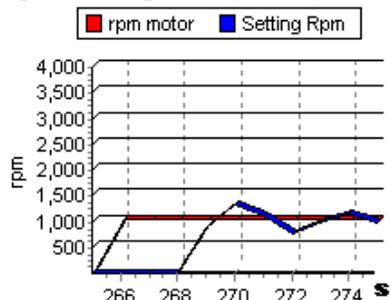
Gambar 8 Respon motor untuk kecepatan 100 rpm

- Gambar respon dari kecepatan motor pada setting kecepatan 500 rpm ditunjukkan gambar 9 .



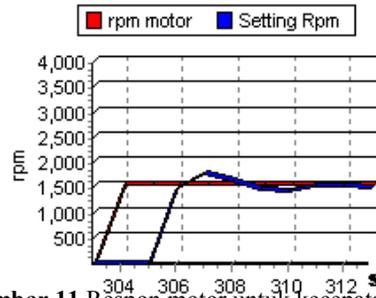
Gambar 9 Respon motor untuk kecepatan 500 rpm

- Gambar respon dari kecepatan motor pada setting kecepatan 1000 rpm ditunjukkan gambar10.



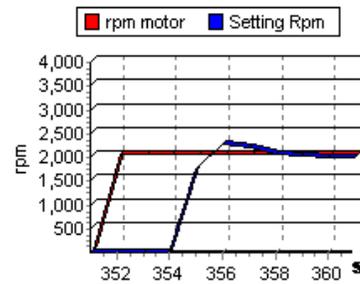
Gambar 10 Respon motor untuk kecepatan 1000 rpm

- Gambar respon dari kecepatan motor pada setting kecepatan 1500 rpm ditunjukkan gambar 11.



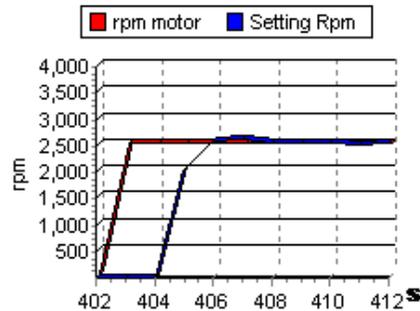
Gambar 11 Respon motor untuk kecepatan 1500 rpm

- Gambar respon dari kecepatan motor pada setting kecepatan 2000 rpm ditunjukkan gambar 12.



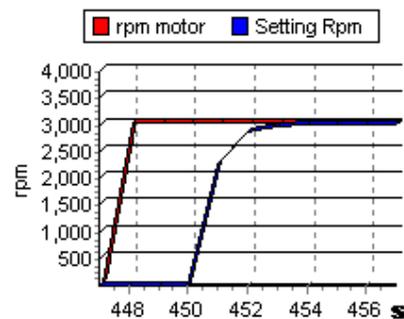
Gambar 12 Respon motor untuk kecepatan 2000 rpm

- Gambar respon dari kecepatan motor pada setting kecepatan 2500 rpm ditunjukkan gambar 13.



Gambar 13 Respon motor untuk kecepatan 2500 rpm

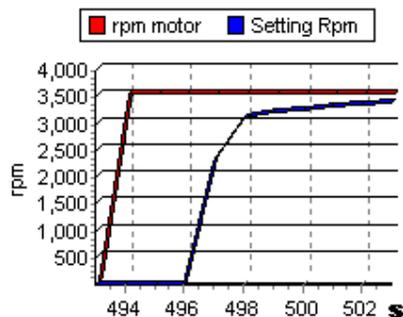
- Gambar respon dari kecepatan motor pada setting kecepatan 3000 rpm ditunjukkan gambar 14.



Gambar 14 Respon motor untuk kecepatan 3000 rpm

- Gambar respon dari kecepatan motor pada setting kecepatan 3500 rpm ditunjukkan gambar 15

Dari berbagai kecepatan setting yang diuji terdapat error *steady state*. Tabel 3 menunjukkan rata-rata error kecepatan yang terjadi dari berbagai pengujian *setting* kecepatan, rata-rata error diambil dari 10 pengambilan data saat *steady state*.



Gambar 15 Respon motor untuk kecepatan 3500 rpm

Tabel 3 rata-rata error steady state dari berbagai setting kecepatan

Setting kecepatan (rpm)	Rata-rata error (rpm)
100	33,4
500	252,2
1000	24,6
1500	21,2
2000	14,4
2500	10,2
3000	8,6
3500	4,2

4. KESIMPULAN

Kesimpulan yang diperoleh dalam perancangan dan pembuatan sistem ini adalah:

- Dalam komunikasi menggunakan TCP/IP data yang dikirim dengan data yang diterima akan sama (error 0 %) karena TCP/IP adalah komunikasi *connection oriented* yaitu komunikasi yang selalu memeriksa komunikasi yang terjadi dan memeriksa data yang ditransmisikan dengan menggunakan *error detection*.
- Agar dapat berkomunikasi dalam jaringan suatu alat harus bisa dikenali dalam jaringan.
- Dalam penelitian ini media transmisi tidak mempengaruhi kesesuaian data yang dikirim dengan data yang diterima
- Untuk mendapatkan respon kecepatan motor yang cepat dengan error yang minimum harus digunakan suatu kontroler *close loop*

REFERENSI

- [1] MCS Electronics Team, *Easy TCP/IP guide*, MCS Electronics, www.mcselec.com.
- [2] D. Ardiyansah, *Teknologi Jaringan Komputer*, www.IlmuKomputer.com
- [3] W.Kelik, *Pengantar Pengkabelan dan Jaringan*, www.IlmuKomputer.com
- [4] T.P.M. Hutapea, *Pengantar Konsep dan Aplikasi TCP/IP Pada Windows NT Server*, www.IlmuKomputer.com
- [5] ATMEL CORPORATION, *Microcontroller with 16K Bytes In-System Programmable Flash: ATMEGA8535*, 2003.
- [6] Tim Wahana Komputer Semarang, *Penanganan Jaringan Komputer*, ANDI OFFSET, 2001
- [7] L. Hendrawan dan B. Wijaya, *Implementation of Embedded Web Server for Remote Control and Monitoring Based on AVR ATMEGA16*, International Conference on Instrumentation, Communication and Information Technology (ICICI) 2005 Proc., August 3 -5, 2005, Bandung, Indonesia
- [8] Wiznet INC, *W3100A Datasheet*, Wiznet Inc, www.WIZnet.co.kr
- [9] D. Purwanto, *Electronic Control System*, Jurusan Teknik Elektro ITS
- [10] ---. *Protokol TCP/IP*, Prasimax, Teknologi Departemen Center, 2002

RIWAYAT HIDUP

Azwardi lahir di Sekayu, MUBA, riwayat pendidikan penulis adalah SDN5 Sekayu, SLTPN 1 Sekayu, SMAN 3 Palembang, diterima di Jurusan Teknik Elektro Politeknik Unsri tahun 1989 menamatkan SI di Teknik Elektro Unsri tahun 1997, Tahun 2012 lulus S2 dari Teknik Elektro ITS bidang studi elektronika. Saat ini penulis adalah tenaga pengajar aktif di Politeknik Negeri Sriwijaya Palembang

