

# RANCANG BANGUN *DRIVER* MOTOR DC SERVO BERBASIS KOMUNIKASI SERIAL PADA ROBOT *HUMANOID* KRISI

**Ronny Ari Setiawan, Nanang Sulistiyanto, Akhmad Zainuri**

Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya

Jalan MT. Haryono 167, Malang 65145, Indonesia

Email : ronny.ari19@gmail.com

## ABSTRAK

Pada umumnya terdapat dua jenis *actuator* yang digunakan pada robot *Humanoid* yaitu motor DC *servo* yang dikontrol secara paralel dan secara serial. Motor DC *servo* paralel memiliki torsi yang cukup besar namun membutuhkan banyak pin mikrokontroler untuk mengendalikannya dalam jumlah banyak, sedangkan motor DC *servo* serial dapat dikontrol hanya dengan satu pin mikrokontroler, namun memiliki torsi yang kecil. Berdasarkan permasalahan tersebut maka dalam penelitian ini dirancangnya sebuah kontroler untuk motor DC *servo* paralel agar dapat dikontrol secara serial. Berdasarkan hasil penelitian, rangkaian *Serial Driver* dapat dirancang menggunakan fitur UART mikrokontroler ATmega8 dengan *baudrate* 1Mbps dan *clock* eksternal 16 MHz. Data dapat ditransmisikan secara efektif dengan metode paket data yang terdiri atas header, ID dan data dalam satu kali pengiriman. Waktu tunggu dalam satu kali pengiriman mencapai 150 $\mu$ s tiap 1 *Serial Driver* dari 6 *Serial Driver* dan data dapat diterima secara akurat oleh rangkaian *Serial Driver*. Sinyal PWM yang dibangkitkan memiliki selisih antara perancangan dan pengukuran sebesar 5  $\mu$ s. Motor DC *Servo* yang digunakan dapat dikategorikan menjadi 2 jenis yaitu motor DC *servo* ideal dan tidak ideal. Motor DC *servo* ideal memiliki selisih sudut antara perancangan dan pengukuran sebesar  $< 4^\circ$ , sedangkan motor DC *servo* tidak ideal memiliki selisih sudut  $> 4^\circ$ . Semakin banyak jumlah tahap yang digunakan untuk menggerakkan motor DC *servo* pada sudut target maka gerakannya semakin halus.

Kata Kunci : DC *servo*, *Serial Driver*,

## ABSTRACT

Generally, there are two kind of *actuator* which is used by *humanoid robot*. They are Parallel DC *servo motor* and Serial DC *servo motor*. The parallel dc *servo motor* has bigger torque than serial DC *servo motor* but it need more than one pin of microcontroller to be controlled. The serial DC *servo motor* can be controlled by one pin but it has smaller torque. Based on that problem, *Serial Driver* has programmed in order to control parallel DC *servo motor* serially. Based on the research, *Serial Driver* can be programmed using UART of microcontroller ATmega8, the *baudrate* is 1 Mbps and using 16 MHz external clock. Data can be transmitted effectivelly using a packet of data which is consist of header, ID and data on one transmission. Delay time each *Serial Driver* of 6 *Serial Driver* reach 150  $\mu$ s and data can be received by *Serial Driver* exactly. PWM signal can be generated with difference value of both PWM signal teory and practice are 5  $\mu$ s. DC *Servo Motor* that used can be classified to 2 kind, they are the ideal and non-ideal DC *servo motor*. The ideal DC *servo motor* has difference value of both teory and practice are less than 4 degree, while non-ideal DC *servo motor* has difference more than 4 degree. If DC *servo motor* uses much number of step to control it, then its motion is soft

**Keywords** : DC *Servo*, *Serial Driver*

## PENDAHULUAN

Kontes Robot Seni Indonesia merupakan suatu kompetisi robotika dimana robot yang dirancang dan dibuat dapat melakukan gerakan tari daerah yang berkembang dan terkenal di wilayah Indonesia. Setiap tim diharuskan membuat sendiri robot otomatis berbentuk *humanoid* yang mampu melakukan gerakan tari mengikuti alunan musik pengiring saat lomba berlangsung. Beberapa poin penilaian yang dilakukan dewan juri berdasarkan panduan Kontes Robot Seni

Indonesia tahun 2015 meliputi kepiawaian gerakan tari di setiap zona, sinkronisasi gerakan dengan musik, dan keatraktifan gerakan serta keluwesan robot dalam melakukan tari. Robot dapat dirancang menggunakan *aktuator* gerak berupa elektromotor, sistem *pneumatic* maupun sistem hidrolik [1].

*Aktuator* gerak jenis elektromotor yang sering digunakan pada robot *humanoid* terdapat dua jenis yaitu motor DC *servo* yang dikontrol secara paralel dan motor DC *servo*

yang dikontrol secara serial. Kedua jenis motor DC servo tersebut memiliki karakteristik masing-masing. Motor DC *servo* paralel memiliki torsi 4,3 kg.cm dan kecepatan 0,13 s/60° sedangkan motor DC *servo* seri memiliki torsi 2 kg.cm dan kecepatan 0,269 s/60° [3].

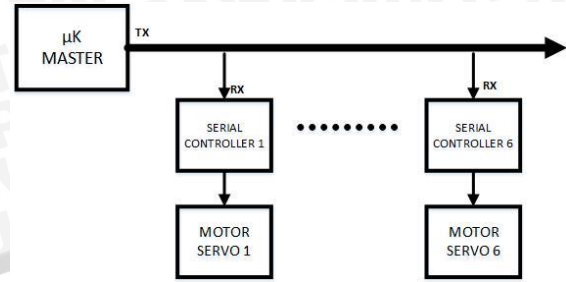
Jenis aktuator yang digunakan oleh tim Teknik Elektro Universitas Brawijaya adalah motor servo paralel. Hal ini dikarenakan robot yang dibuat memiliki massa cukup besar sehingga membutuhkan aktuator dengan torsi yang besar. Namun, penggunaan motor servo paralel memiliki beberapa kerugian yaitu membutuhkan banyak pin mikrokontroler untuk mengontrol lebih dari satu motor DC servo karena setiap motor DC servo paralel membutuhkan satu pin mikrokontroler, membutuhkan kabel yang cukup panjang untuk menjangkau sendi terjauh dari mikrokontroler.

Berdasarkan masalah-masalah tersebut maka dalam skripsi ini penulis merancang kontroler untuk motor DC servo paralel agar dapat dikontrol secara serial. Dengan menggunakan sistem ini diharapkan instalasi kabel pada robot menjadi lebih mudah, simpel dan rapi serta gerakan robot menjadi lebih luwes.

## METODE PENELITIAN

### A. Diagram Blok Sistem

Metode yang digunakan pada penelitian ini yaitu menghubungkan beberapa mikrokontroler menggunakan komunikasi serial *multipoint*. Metode ini memungkinkan untuk menghubungkan banyak divais secara serial [2]. Diagram blok perancangan untuk keseluruhan sistem ditunjukkan pada Gambar 1.



Gambar 1. Diagram blok keseluruhan sistem.

Seperti ditunjukkan pada gambar 1, keseluruhan sistem pada penelitian ini terdiri atas 1 buah blok *master* dan 6 buah blok *slave* yang terhubung secara seri. Keenam blok *slave* memiliki struktur yang sama, yang membedakan antara satu *slave* dengan *slave* lainnya yaitu ID yang ditanamkan untuk masing-masing *slave*. Penjelasan untuk masing-masing blok yaitu sebagai berikut:

- Master akan mengirim paket data berupa ID *slave* dan nilai untuk menggerakkan motor DC *servo*
- Data dengan ID yang sesuai dengan *Serial Driver* akan dimasukkan pada variabel untuk membangkitkan sinyal PWM penggerak motor DC *servo*. Jika tidak ada ID yang sesuai dengan *Serial Driver* maka variabel untuk membangkitkan sinyal PWM diisi dengan nilai sebelumnya.

Adapun diagram untuk blok *slave* ditunjukkan pada Gambar 2.



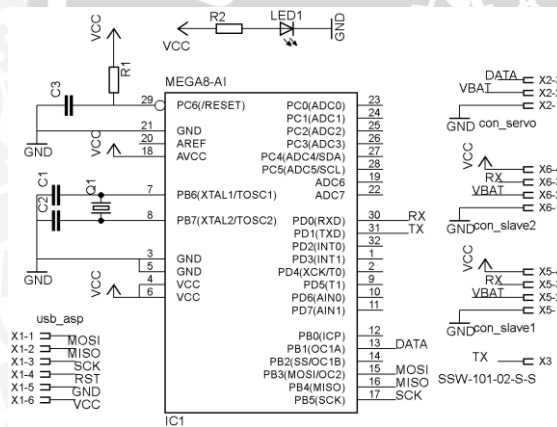
Gambar 2. Diagram blok *Serial Driver* .

Penjelasan untuk cara kerja blok *slave Serial Driver* yaitu, pertama mikrokontroler menunggu apakah ada paket data yang diterima melalui pin RX. Paket data tersebut berisi ID dan nilai sudut. Paket data yang diterima kemudian diperiksa apakah memiliki ID yang sama. Jika paket data tersebut memiliki ID yang sama maka nilai sudut yang

mengikutinya akan dikonversi menjadi nilai tertentu dan dimasukkan pada variabel yang mengatur lebar sinyal high PWM. Sinyal PWM yang dibangkitkan akan digunakan untuk menggerakkan motor DC *servo*. Jika ID-nya berbeda maka variabel yang mengatur lebar sinyal high PWM akan diisi nilai sebelumnya.

**B. Perancangan Hardware**

Rangkaian *mainboard* ATmega8 merupakan rangkaian utama sistem *Serial Driver* pada penelitian ini. Rangkaian ini berupa *minimum system* ATmega8 yang dilengkapi dengan beberapa fitur yaitu 1 pin *Timer/Counter1* untuk antarmuka motor DC *servo*, 1 pin *receiver* antarmuka USART, 6 pin untuk antarmuka USB ASP programmer serta konektor catu daya. Gambar 3 menunjukkan rangkaian mainboard ATmega8.



Gambar 3. Rangkaian mainboard Serial Driver .

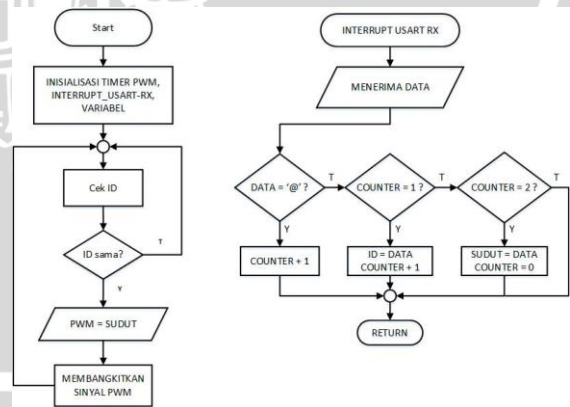
Pada rangkaian mainboard ATmega8 terdapat dua buah catu daya yaitu catu daya untuk motor DC servo dan catu daya untuk mikrokontroler. Catu daya untuk motor DC servo menggunakan tegangan 6 V dan untuk mikrokontroler menggunakan tegangan 5 V.

Sumber *clock* yang digunakan yaitu *crystal* dengan frekuensi 16 MHz. Dua kapasitor non polar yang dihubungkan secara paralel pada kedua ujung kaki *crystal* memiliki kapasitansi sebesar 22 pF. Resistor *pull up* yang menghubungkan antara pin reset dengan VCC memiliki resistansi sebesar 4,7

kΩ. Kapasitor non polar yang menghubungkan antara pin reset dengan *ground* memiliki kapasitansi sebesar 100 nF. Led yang dihubungkan dengan sumber tegangan digunakan sebagai indikator apakah terdapat tegangan yang menyuplai rangkaian. Agar led tidak rusak maka perlu diberi resistor untuk menghambat arus yang melewati led. Berdasarkan datasheet arus maksimum pada led yaitu 20 mA, sehingga dengan menggunakan hukum ohm diperoleh nilai resistor sebesar 250 Ω. Karena resistor 250 Ω tidak tersedia di pasaran maka digunakan resistor 270 Ω. Jika nilai resistor 270 Ω dimasukkan kembali maka akan diketahui arus yang melewati led yaitu 18 mA.

**C. Perancangan Software**

Perancangan *Software* dilakukan dengan merancang diagram alir (*flowchart*) terlebih dahulu. Diagram alir ini berfungsi sebagai alur kerja untuk setiap perangkat keras yang dikendalikan oleh mikrokontroler ATmega8 serta proses-proses perhitungan yang dikerjakan oleh mikrokontroler. Bahasa pemrograman yang digunakan dalam program utama adalah bahasa C dengan menggunakan *compiler* AVR Studio 7.0. Diagram alir perancangan *software* ditunjukkan pada Gambar 4.



Gambar 4. Diagram alir software.

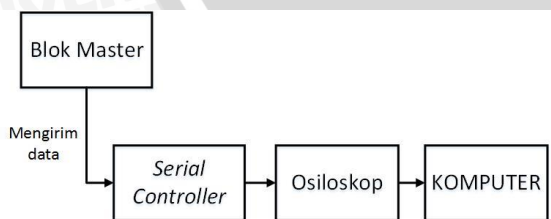
Proses yang pertama kali dilakukan adalah inisialisasi *Timer/Counter1*, *interrupt uart receiver*, dan variabel-variabel yang digunakan. *Timer/Counter1* diatur pada mode *fast PWM* untuk mengendalikan motor DC

*servo*. *Interrupt* usart *receiver* digunakan untuk menginterupsi program ketika terdapat data yang diterima melalui pin RX ATmega8. Setelah inisialisasi, proses selanjutnya yaitu menunggu apakah terdapat data yang diterima melalui pin RX, apabila ada maka sub rutin *interrupt* usart akan menginterupsi program utama. Data yang diterima berupa paket data sederhana dengan susunan karakter '@' - data Id - data nilai sudut. Untuk memisahkan paket data tersebut dilakukan proses pemilihan kondisi. Apabila data yang diterima berupa karakter '@' maka variabel counter ditambah 1. Apabila data yang diterima bukan karakter '@' dan variabel counter bernilai 1 maka data akan dimasukkan pada variabel ID dan variabel counter ditambah 1. Apabila data yang diterima bukan karakter '@' dan variabel counter bernilai 2 maka data akan dimasukkan pada variabel Sudut dan variabel counter diberi nilai 0. Setelah didapatkan data Id dan nilai sudut, proses selanjutnya yaitu memeriksa data pada variabel ID apakah memiliki Id yang sesuai dengan *Serial Driver*. Jika sesuai maka data pada variabel Sudut akan dikonversi menjadi nilai digital untuk membangkitkan sinyal pwm dan menggerakkan motor DC *servo*.

## HASIL DAN PEMBAHASAN

### A. Pengujian Transmisi Data

Pengujian ini dilakukan untuk mengetahui keberhasilan transmisi data dari blok *master* ke rangkaian *Serial Driver*. Alat bantu yang digunakan yaitu osiloskop. Skema pengujian ditunjukkan pada Gambar 5.



Gambar 5. Skema pengujian transmisi data.

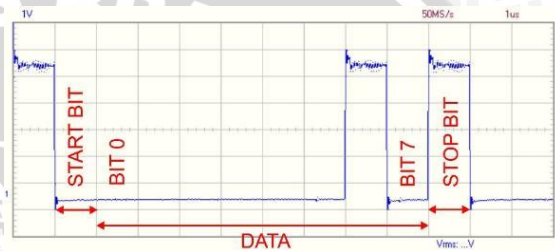
Blok master mengirimkan beberapa karakter dengan *baudrate* 1 Mbps. Setiap karakter yang dikirim memiliki format *frame* seperti ditunjukkan pada Gambar 6.

St	0	1	2	3	4	5	6	7	Sp
----	---	---	---	---	---	---	---	---	----

Gambar 6. Format *frame* pengiriman data.

St merupakan bit *Start*, 0-7 merupakan bit data dan Sp merupakan bit *Stop*. Bit *start* selalu berlogika *high* dan bit *stop* selalu berlogika *low*.

Data yang diterima rangkaian *Serial Driver* kemudian dikirim kembali melalui pin tx agar dapat dibaca melalui osiloskop. Salah satu contoh hasil pembacaan osiloskop karakter '@' ditunjukkan pada Gambar 7.



Gambar 7. Hasil pembacaan data dengan osiloskop.

Berdasarkan hasil pengujian, data yang dikirim dari blok *master* dapat diterima secara akurat oleh rangkaian *Serial Driver*. Hasil pengujian ditunjukkan pada Tabel 1.

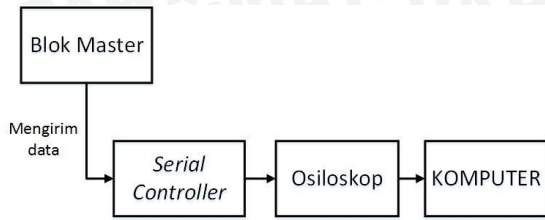
Tabel 1. Hasil Pengujian Transmisi Data

DATA KIRIM (DEC)	DATA TERIMA (BINER)									
	St	0	1	2	3	4	5	6	7	Sp
1	1	1	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	0	0
4	1	0	0	1	0	0	0	0	0	0
5	1	1	0	1	0	0	0	0	0	0
6	1	0	1	1	0	0	0	0	0	0
10	1	0	1	0	1	0	0	0	0	0

### B. Pengujian Sinyal PWM

Pengujian ini bertujuan untuk mengetahui pengaruh perubahan nilai data yang diterima melalui komunikasi UART terhadap lebar  $T_{ON}$  pada sinyal PWM penggerak motor DC *servo*. Alat bantu yang digunakan pada pengujian ini yaitu osiloskop digital dan komputer. Osiloskop digital berfungsi untuk

membaca bentuk sinyal PWM yang dibangkitkan oleh rangkaian *Serial Driver* dan komputer digunakan untuk menampilkan hasil pembacaan osiloskop. Skema pengujian sinyal PWM ditunjukkan pada Gambar 8.



Gambar 8. Skema pengujian sinyal PWM.

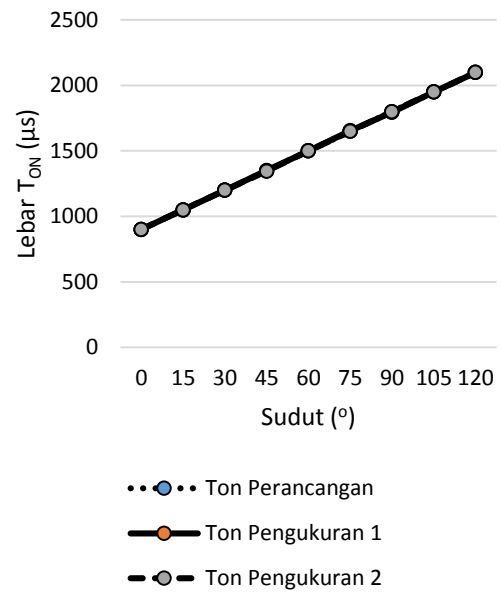
Pertama blok *master* mengirimkan data yang berisi nilai sudut dengan rentang  $0^\circ - 120^\circ$ . Nilai sudut yang diterima rangkaian *Serial Driver* kemudian dikonversi menjadi nilai pembangkit sinyal PWM. Sinyal PWM dikeluarkan pada PORTC.0 mikrokontroler ATmega8 untuk dilihat bentuk sinyalnya menggunakan osiloskop. Osiloskop diatur pada bagian volt/div dan time/div dengan pengaturan 1 V/div dan 1 ms/div. Hasil pengujian sinyal PWM ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pengujian Sinyal PWM

Data Terima ( $^\circ$ )	$T_{ON}$ Perancangan ( $\mu s$ )	$T_{ON}$ Pengukuran ( $\mu s$ )			
		SD1	SD2	SD3	SD4
0	900	900	900	900	900
15	1050	1050	1050	1055	1055
30	1200	1200	1200	1200	1200
45	1350	1350	1345	1345	1345
60	1500	1500	1500	1500	1500
75	1650	1655	1650	1650	1655
90	1800	1795	1800	1800	1800
105	1950	1950	1950	1950	1950
120	2100	2100	2100	2100	2100

\*SD = *Serial Driver*

Berdasarkan tabel diatas, masing – masing rangkaian *Serial Driver* dapat menghasilkan lebar  $T_{ON}$  sinyal PWM yang sama dengan perancangan. Namun terdapat beberapa selisih yang terjadi pada  $T_{ON}$  pengukuran, selisih tersebut yaitu sebesar 5 angka. Jika dirubah kedalam bentuk grafik, hasil pengujian ditunjukkan pada Gambar 9.

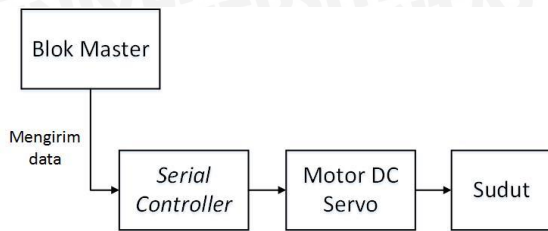


Gambar 9. Grafik pengujian sinyal PWM.

Seperti yang ditunjukkan pada grafik, dapat diambil kesimpulan bahwa grafik lebar  $T_{ON}$  berupa kurva linear dengan data perancangan dan pengukuran yang berimpit. Selisih antara lebar  $T_{ON}$  perancangan dan pengukuran yaitu sebesar 0-5. Namun selisih tersebut tidak berpengaruh pada sudut yang akan dihasilkan motor DC *servo* karena perubahan sudut  $1^\circ$  terjadi setiap selisih 10  $\mu s$ .

### C. Pengujian Motor DC *Servo*

Pengujian ini dilakukan untuk mengetahui apakah motor DC *servo* dapat bergerak pada sudut tertentu sesuai dengan data yang dikirim oleh blok *master*. Sumber tegangan yang digunakan pada pengujian ini berasal dari baterai *lithium polymer* 2S 7,4 V 2000 mAh. Tegangan dari baterai kemudian diturunkan menjadi 6 V menggunakan rangkaian *regulator LM2576 simple switcher* untuk menyuplai motor DC *servo* dan untuk menyuplai rangkaian *Serial Driver* digunakan rangkaian *linear regulator L7805*. Alat bantu yang digunakan pada pengujian ini yaitu busur derajat yang berfungsi sebagai penunjuk sudut pergerakan motor DC *servo*. Skema pengujian ditunjukkan pada Gambar 10.



Gambar 10. Skema pengujian motor DC servo.

Prosedur yang dilakukan pada pengujian ini sama dengan pengujian sinyal PWM, namun keluaran sinyal PWM disambungkan langsung dengan motor DC *servo* untuk mengamati sudut pergerakannya. Busur derajat dipasang pada *horn* motor DC *servo* dan disesuaikan pada sudut 0°. Pengaturan sudut 0° diwakilkan pada lebar  $T_{ON}$  900  $\mu$ s. Data hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3. Hasil Pengujian Motor DC Servo

$T_{ON}$ ( $\mu$ s)	Sudut Perancangan (°)	Sudut Pengukuran (°)			
		MS1	MS2	MS3	MS4
900	0	0	0	0	0
1100	20	20	23,5	21,5	21,5
1200	30	30,5	35,5	32	32
1300	40	40,5	46,5	41,5	42
1500	60	60	68,5	61,5	61
1700	80	81	90	81,5	81
1900	100	101,5	111	101	101
2100	120	123,5	134	121,5	119,5

\*MS = Motor DC Servo

Berdasarkan tabel hasil pengujian, terdapat motor DC *servo* yang ideal sesuai dengan perancangan dan motor DC *servo* yang tidak ideal. Motor DC *servo* yang ideal memiliki selisih pengukuran tertinggi sebesar 4° dan selisih terkecil 0,5°. Motor DC *servo* yang tidak ideal memiliki selisih pengukuran diatas 4°. Motor DC *servo* yang ideal seharusnya dapat bergerak dari 0° – 120° pada rentang lebar  $T_{ON}$  900 – 2100  $\mu$ s, namun untuk motor DC *servo* yang tidak ideal bergerak dari 0° – 134° pada rentang lebar  $T_{ON}$  yang sama.

#### D. Pengujian Keseluruhan Sistem

Pengujian ini bertujuan untuk membuktikan bahwa tiap blok yang telah diuji dapat dirangkai menjadi satu sistem yang utuh dan dapat bekerja sesuai dengan perencanaan. Pengujian sudut dilakukan sebanyak 2 kali dengan memberikan sudut 0°-120° pada setiap

motor DC *servo* dengan 2 variasi waktu yaitu satu kali tahap dengan jeda 50 ms untuk mencapai sudut target dan 20 kali tahap dengan jeda 50 ms untuk mencapai sudut target. Data hasil pengujian ditunjukkan pada Tabel 4.

Tabel 4. Hasil Pengujian Keseluruhan

Perc. ke	Sudut Perancangan (°)	Sudut Pengukuran (°)			
		MS1	MS2	MS3	MS4
1	0	0	0	0	0
2	20	22	19	22	18
3	40	43	41	42	40
4	60	61	61	62	57
5	80	80	81	72	76
6	100	102	103	96	96
7	120	123	123	121	118

1 step

Perc. ke	Sudut Perancangan (°)	Sudut Pengukuran (°)			
		MS1	MS2	MS3	MS4
1	0	0	0	0	0
2	20	22	21	22,5	21,0
3	40	44	42	44	42,5
4	60	63	61	64	63
5	80	82	82	83	83,5
6	100	104	102	103	104
7	120	124	124	124	123

20 step

Berdasarkan data hasil pengujian pada Tabel 4, selisih antara sudut perancangan dan pengukuran pada motor DC *servo* 1 dengan 1 tahap memiliki rata-rata sebesar 1,57°, pada motor DC *servo* 2 sebesar 1,43°, pada motor DC *servo* 3 sebesar 2,71°, pada motor DC *servo* 4 sebesar 2,14°. Sedangkan selisih antara sudut perancangan dan pengukuran pada motor DC *servo* 1 dengan 20 tahap memiliki rata-rata sebesar 2,71°, pada motor DC *servo* 2 sebesar 1,71°, pada motor DC *servo* 3 sebesar 2,93°, pada motor DC *servo* 4 sebesar 2,43°.

Motor DC *servo* dengan 20 tahap memiliki pergerakan lebih halus namun menghasilkan selisih rata-rata lebih besar jika dibandingkan dengan pergerakan motor DC *servo* dengan 1 tahap. Selisih tersebut terjadi karena adanya penambahan dan pembulatan

hasil pembagian tahap yang digunakan. Namun selisih yang dihasilkan dapat diabaikan karena dalam Kontes Robot Seni Indonesia yang menjadi poin penting adalah keluwesan dan kehalusan gerakan robot.

## KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan rumusan masalah, perancangan dan hasil pengujian maka dapat diambil kesimpulan sebagai berikut

1. Motor DC *servo* paralel dapat dikontrol secara serial menggunakan metode komunikasi serial *multipoint*, komunikasi *simplex* dan pengiriman data menggunakan metode paket data yang terdiri dari 1 *byte Header*, ID dan Sudut.
2. Rangkaian *Serial Driver* dapat dirancang menggunakan *peripheral* UART mikrokontroler ATmega8 dengan *Baudrate* 1 Mbps dan PWM dengan resolusi 16 bit.
3. Rangkaian *Serial Driver* mampu menerima data secara akurat dengan waktu tunggu 150  $\mu$ s tiap *Serial Driver* dari 6 *Serial Driver*, mampu membangkitkan sinyal PWM dengan selisih antara perancangan dan pengukuran 0-5  $\mu$ s, mampu menggerakkan motor DC *servo* dengan selisih terbesar 4° untuk motor DC *servo* ideal dan selisih diatas 4° untuk motor DC *servo* tidak ideal. Semakin banyak jumlah tahap yang digunakan untuk mencapai sudut target maka gerakan motor DC *servo* semakin halus.

### B. Saran

Penelitian lebih lanjut mengenai *Serial Driver* untuk motor DC *servo* perlu dilakukan penyempurnaan pada algoritma tahap agar tidak terjadi pembulatan disetiap hasil bagi jumlah tahap yang digunakan sehingga pergerakan motor DC *servo* menjadi lebih akurat.

## DAFTAR PUSTAKA

- [1] Dikti. 2014. *Panduan KRSI 2015-ver Okt 2014*. Jakarta: Direktorat Jenderal Pendidikan Tinggi Kementrian Riset Teknologi dan Pendidikan Tinggi
- [2] Jan Axelson, *Networks for Monitoring And Control Using an RS485 Interface*, Mikrocomputer Journal, August 1995
- [3] Lee Jun Hee. 2007. *General Specification of HS-5085MG Digital Micro Servo*. Korea: Hitec Rcd Korea Inc