

# PENGENDALI MOTOR SERVO BERBASIS MIKROKONTROLER BASIC STAMP 2SX UNTUK MENGEMBANGKAN SISTEM ROBOTIKA

**Sujarwata**

Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Negeri Semarang (UNNES) Semarang  
e-mail: sjarwot@yahoo.co.id

## Abstrak

Motor Servo merupakan salah satu jenis aktuator yang cukup banyak digunakan dalam bidang industri atau sistem robotika. Motor servo yang digunakan dalam penelitian ini adalah servo FP-S3001 dengan putaran cepat dan dapat dikendalikan. Sebelum digunakan motor servo harus dimodifikasi terlebih dahulu, hal ini dikarenakan standar pabrik putaran servo hanya mencapai  $180^\circ$ . Oleh sebab itu motor servo harus dimodifikasi agar dapat mencapai putaran  $360^\circ$  (satu putaran penuh).

Pengendalian putaran motor servo dilakukan dengan menggunakan metode PWM. (Pulse Width Modulation). Pengendali motor servo digunakan BASIC Stamp2SX dengan bahasa pemrograman PBasic yang sangat sederhana. Hasil penelitian didapatkan, bahwa: putaran motor servo searah dengan arah jarum jam, jika jumlah pulsa yang harus di-input-kan lebih kecil dari 1600 pulsa (1,6 ms pulsa). Sedangkan motor servo akan berputar berlawanan dengan arah jarum jam, jika jumlah pulsa yang di-input-kan lebih besar dari 1600 pulsa (1,6 ms pulsa).

**Kata kunci:** BASIC Stamp2SX, Motor Servo, PBasic, PWM

## Abstract

*The servo motors are one type of actuator that is widely used in industry or robotics systems. The servo motors are used in this study is FP-S3001 The servo with fast laps and can be controlled. Prior to use The servo motors to be modified first, this is because the standard factory round The servo only reach  $180^\circ$ . Therefore The servo motors must be modified in order to reach the round of  $360^\circ$  (one full turn).*

*The servo motor control is done by using the PWM method. (Pulse Width Modulation). Servo motor controller used with the programming language BASIC Stamp2SX PBasic is very simple. The results showed that: The servo motor rotation direction clockwise, if the number of pulses that must be input-it is smaller than 1600 pulses (1.6 ms pulse). While the servo motor will rotate counter-clockwise, if the number of pulses that are input-kan greater than 1600 pulses (1.6 ms pulse).*

**Keywords :** BASIC Stamp2SX, Motor Servo, PBasic, PWM

## 1. Pendahuluan

Dunia robotika saat ini berkembang sangat pesat, simak saja robot Asimo produksi Honda yang dapat bergerak menyerupai manusia. Untuk mengendalikan robot dengan *microcontroller* sebagai otak robot. Pengembangan sistem robotika di dalam dan di luar negeri, kami justru melihat bahwa penelitian mengenai sistem robotik ini sangat penting dan perlu di kaji.. Pada sisi lain kurangnya penelitian mengenai robot yang memadai dan pengembangan penelitian lebih lanjut. Sebagai peneliti yang tertarik pada bidang robot,

maka perlu ada pihak yang memulai mengembangkan atau mengangkat topik tentang sistem robotika.

Masalah yang muncul pada sistem robotika ini adalah bagaimana cara mengendalikan dan memasang penggerak (motor) yang tepat, sehingga robot dapat melakukan gerakan sesuai dengan perintah. Pemasangan motor sebagai penggerak robot memerlukan teknik tersendiri dengan memperhatikan karakteristik, posisi dan cara mengendalikan. Sebagai contoh memasang motor pada robot harus mengetahui spesifikasi dan bagaimana cara mengendalikannya (sistem otomasi). Dengan memahami berbagai macam motor dan mikrokontroler untuk sistem otomasi pengaturannya, maka mahasiswa akan lebih mudah dalam mempelajari sistem robotika. Disamping itu dalam memilih jenis mikrokontroler yang akan digunakan sering kurang tepat, sehingga mengalami kesulitan dalam menyusun program pengendalinya. Untuk mengatasi hal demikian dapat digunakan mikrokontroler Basic Stamp dengan bahasa pemrograman Pbasic yang relatif sederhana, bahkan dapat menghindari algoritma rumit. Mikrokontroler ini keluarannya sudah dalam bentuk data digital, sehingga tidak lagi memerlukan *interface*. Dalam mikrokontroler juga dilengkapi dengan *intepreter* yang secara otomatis menunjuk register tertentu, dengan demikian akan memudahkan pengguna. Perintah dengan bahasa sederhana dan ditulis langsung ke bentuk desimal, tidak perlu mengubahnya ke bentuk biner.

Disamping itu dalam memilih jenis motor dan mikrokontroler yang akan digunakan sering pada robot kurang tepat, sehingga mengalami kesulitan dalam mengerakkan dan menyusun program pengendalinya. Untuk mengatasi hal demikian dapat digunakan servo motor dan mikrokontroler *Basic Stamp 2SX*. Mikrokontroler *Basic Stamp 2SX* dengan bahasa pemrograman Pbasic yang relatif sederhana, bahkan dapat menghindari algoritma rumit. Mikrokontroler ini keluarannya sudah dalam bentuk data digital, sehingga tidak lagi memerlukan *interface*. Dalam mikrokontroler juga dilengkapi dengan *intepreter* yang secara otomatis menunjuk register tertentu, dengan demikian akan memudahkan pengguna. Perintah dengan bahasa sederhana dan ditulis langsung ke bentuk desimal, tidak perlu mengubahnya ke bentuk biner.

## 2. Tinjauan Pustaka

Sebuah motor servo *standard* adalah alat yang dapat mengendalikan posisi, dapat membelokkan dan menjaga suatu posisi berdasar penerimaan pada suatu signal elektronik itu. Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnet permanent motor DC servo yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet.

Bagian - bagian dari sebuah motor servo *standard* adalah sebagai berikut:

- a) Konektor yang digunakan untuk menghubungkan motor servo dengan *Vcc*, *Ground* dan signal *input* yang dihubungkan ke *Basic Stamp*.
- b) Kabel menghubungkan *Vcc*, *Ground* dan signal *input* dari konektor ke motor servo.
- c) Tuas menjadi bagian dari motor *servo* yang kelihatan seperti suatu bintang *four-pointed*. Ketika motor servo berputar, tuas motor servo akan bergerak ke bagian yang dikendalikan sesuai dengan program.
- d) Casing berisi bagian untuk mengendalikan kerja motor servo yang pada dasarnya berupa motor DC dan *gear*. Bagian ini bekerja untuk menerima instruksi dari *basic stamp* dan mengkonversi ke dalam sebuah pulsa untuk menentukan arah / posisi servo.

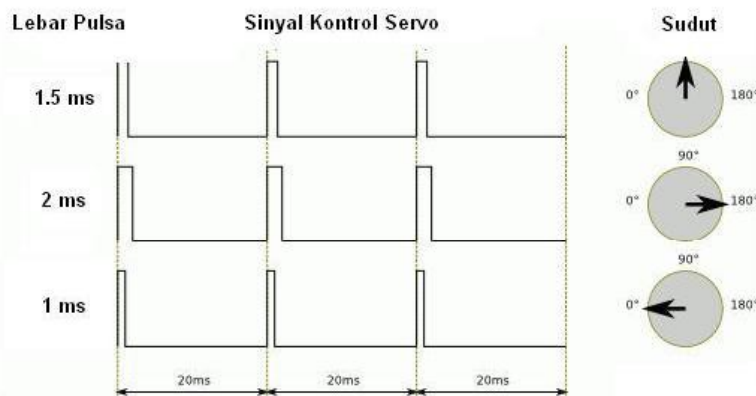
## 2.1 Motor Servo

Motor servo menggunakan dengan sistem umpan balik tertutup, di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energy mekanik, maka magnet permanent motor DC servolah yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet. Salah satu medan dihasilkan oleh magnet permanent dan yang satunya dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnet tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.

## 2.2 Tipe Motor Servo

Secara umum terdapat 2 jenis motor servo. yaitu motor servo standard dan motor servo Continous. Servo motor tipe standar hanya mampu berputar 180 derajat. Motor servo standard sering dipakai pada sistim robotika misalnya untuk membuat “Robot Arm” (Robot Lengan). Sedangkan Servo motor continuous dapat berputar sebesar 360 derajat. motor servo *continous* sering dipakai untuk Mobile Robot. Pada badan servo tertulis tipe servo yang bersangkutan. Pengendalian gerakan batang motor servo dapat dilakukan dengan menggunakan metode PWM. (Pulse Width Modulation). Teknik ini menggunakan system lebar pulsa untuk mengemudikan putaran motor. Sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar dengan pulsa 1.5 ms pada periode selebar 2 ms, maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.

Untuk menggerakkan motor servo ke kanan atau ke kiri, tergantung dari nilai delay yang kita berikan. Untuk membuat servo pada posisi center, berikan pulsa 1.5ms. Untuk memutar servo ke kanan, berikan pulsa  $\leq 1.3\text{ms}$ , dan pulsa  $\geq 1.7\text{ms}$  untuk berputar ke kiri dengan delay 20ms, seperti ilustrasi berikut:



Gambar 1. Pensinyalan motor servo

### 2.3 PWM (*Pulse width Modulation*)

PWM (*Pulse width Modulation*), adalah sebuah metode untuk pengaturan kecepatan perputaran, dalam hal ini adalah motor DC untuk gerak robot. PWM dapat dihasilkan oleh empat metode, sebagai berikut :

- a. Metode analog
- b. Metode digital
- c. IC diskrit
- d. Mikrokontroler

Pada robot ini, metode PWM dikerjakan oleh mikrokontroler. Metode PWM ini akan mengatur lebar atau sempitnya periode pulsa aktif yang dikirimkan oleh mikrokontroler ke driver motor. Pada pengaturan kecepatan robot, nilai PWM mulai dari 0-255. Secara *analog* besaran PWM dihitung dalam prosentase, nilai ini didapat dari perbandingan:  $T_{high} / (T_{high} + T_{low}) * 100\%$ . Dimana T adalah periode atau waktu tempuh untuk sebuah pulsa, yang terbagi menjadi bagian puncak positif (*T<sub>high</sub>*) dan puncak negatif (*T<sub>low</sub>*).

Semakin rapat periode antar pulsa, maka frekuensi yang dihasilkan akan semakin tinggi, ini berarti kecepatan akan bertambah. Semakin lebar jarak antar pulsa, maka frekuensi semakin rendah ini berarti kecepatan berkurang atau menurun. Kondisi pemberian kecepatan harus disesuaikan dengan kondisi *track* yang akan dilewati oleh robot, misal pada saat jalan lurus, naik atau turun harus mendapatkan nilai PWM yang tepat.

Permasalahan dalam penelitian ini adalah sebagai berikut:

- a) Bagaimana mengendalikan motor servo agar putarannya sesuai dengan yang dikehendaki.
- b) Bagaimana teknik memasang motor servo tepat, sehingga dapat bekerja secara efektif.
- c) Bagaimana cara menyusun program sehingga motor servo dapat berputar sesuai dengan perintah.

### 3. Metodologi

Pada riset ini akan digunakan mikrokontroler *Basic Stamp 2SX* sebagai sistem pengendali pada robot. Adapun spesifikasi fungsi kaki Mikrokontroler *Basic Stamp 2SX* adalah sebagai berikut:

- a. Terdiri dari 24 pin dan bekerja pada frekuensi 50 MHz
- b. 16 Kbyte memory EEPROM, terdiri dari 8 section , 2 Kbyte/section
- c. 32 byte RAM dengan 64 byte Scratcpad RAM
- d. Kecepatan eksekusi 10.000 instruksi/detik dan dapat dihapus 100.000 kali
- e. Konsumsi arus 65 mA pada saat operasi penuh, dan 20  $\mu$ A saat tidak beroperasi
- f. Menggunakan PC serial port untuk programming dan debugging
- g. Terdiri 16 pin I/O independent
- h. I/O pin dapat menggerakkan LED dan 6 V servo motor secara langsung
- i. Terdapat LED *power status*
- j. Menggunakan bahasa pemrograman Pbasic (*Parallax Basic*)
- k. *Basic Stamp* terdiri dari *Intepreter*, EEPROM dan Resonator.

### 3.1 Men-download Program ke Mikrokontroler *Basic Stamp 2SX*

Pemrograman Sistem Robotika dengan Basic Stamp2SX. Ketentuan-ketentuan dan penjelasan cara pemrograman sebagai berikut:

- a. Untuk menulis program menggunakan *Basic Stamp Editor* yang sudah di-install sebelumnya pada PC. Setelah menulis program, PC dihubungkan dengan *Basic Stamp Microcontroller*. *Basic Stamp* harus dihubungkan dengan *battery supply*. Tekan *run* pada *Basic Stamp Editor* untuk proses *down-load* program ke *EEPROM*.
- b. Proses *down-load* program otomatis menghapus program yang ada sebelumnya dan mikrokontroler akan mengeksekusi program yang terakhir.
- c. Program ditulis dalam *Basic Stamp Editor* dan disimpan dalam format *\*BS2SX*.
- d. Program dalam bentuk format *\*BS2SX* akan diubah selama proses *down-load* ke dalam format biner yang dapat dibaca oleh mikrokontroler menggunakan interpreter pada *Basic Stamp*. Komentar “*Tokenzied is successfully*” akan muncul bila program *Pbasic* telah berhasil.
- e. Untuk reprogram *Basic Stamp* hanya dengan menghubungkan PC, kemudian dijalankan *Basic Stamp Editor*, selanjutnya tekan Alt-R atau RUN.
- f. Setiap perintah dalam suatu program akan selalu menempati ruang dalam *EEPROM*, besarnya bervariasi sesuai dengan kompleksitas dan jenis perintah yang ditulis.

Motor servo yang digunakan dalam penelitian ini adalah servo FP-S3001 dengan berbagai kemudahan. Adapun karakteristik Servo FP-S3001, sebagai berikut:

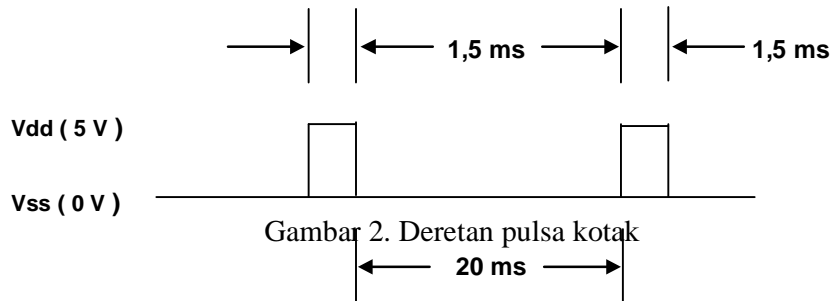
- a) Mempunyai tinggi 39 mm (1,4 inchi) dan dapat digunakan untuk perencanaan berbagai model.
- b) Bola sekrup dapat digunakan untuk output, operasi ketika servo terbebani sangat lembut, getaran dan keretakan telah dicegah dengan berbagai perbaikan.
- c) Getaran dan daya tahan diperbaiki dengan menggunakan petunjuk penulisan sistem, amplifier, motor dan potensiometer servo.
- d) Daya putar tinggi, sehingga menyebabkan putaran servo sangat cepat dan perpindahan putaran servo lembut.
- e) Mengendalikannya tidak langsung, potensiometer lengkap secara tertutup yang digunakan untuk memperbaiki bagian pokok, yaitu: getaran dan perlawanan keretakan serta akuranya tinggi.
- f) Sistem pengendalinya dengan menggunakan lebar pulsa 1520  $\mu\text{m}$  (15,20 ms) dalam kondisi netral dan *power supply* 4,8 volt.

### 3.2 Modifikasi motor servo

Motor servo harus dimodifikasi terlebih dahulu, hal ini dikarenakan standar pabrik putaran servo hanya mencapai  $180^\circ$ . Oleh sebab itu motor servo dimodifikasi agar dapat mencapai putaran  $360^\circ$  (satu putaran penuh). Modifikasi motor servo dengan cara mengganti R variabel (potensiometer) dengan R tetap, yaitu 2,7 k $\Omega$  sebanyak 2 buah dipasang seri dan menghilangkan bagian stop tab pada final *gear servo*. Sedangkan kabel konektor motor servo harus diubah urutan kabelnya, karena untuk menyesuaikan dengan urutan konektor pada pin I/O *main board*.. Ada 3 kabel konektor dengan urutan (“*GND*”), putih untuk *Sources Data* dan merah untuk *Vcc*.

### 3.3 Pemrograman cara kerja motor servo DC

Sistem kendali BASIC Stamp2SX dengan cara mengirimkan sejumlah deretan pulsa kotak ke motor servo, kemudian motor servo merespon. Deretan pulsa kotak mempunyai high time 1,5 ms dan low time 20 ms. *High time* seringkali dikaitkan dengan lebar pulsa positif dan mempunyai tegangan puncak 5 volt. Sedangkan *low time* berkaitan dengan dengan keadaan, dimana pulsa mempunyai mempunyai tegangan 0 volt.



Gambar 2. Deretan pulsa kotak

Pada umumnya dalam menggunakan perintah *PBASIC* harus memulai dengan kata-kata yang ada di daftar tabel secara otomatis. Sebuah kata *loop* merupakan daftar kata bersama dengan perintah *goto loop* pada akhir suatu program. Sebuah kata *loop* juga merupakan daftar petunjuk tempat dan sewaktu-waktu dapat memanggil program dengan perintah, misalnya: “ *loop: label* “, hasilnya adalah *pulsout* dan *pause*. Perintah eksekusi dapat diambil berulang-ulang dengan mengirimkan *signal center* ke servo.

Kata *loop: .... goto* adalah struktur program yang dikatakan sebagai putaran tak terbatas. Putaran tak terbatas merupakan bagian dari *code* dalam program akhir dan dapat dimulai lagi tanpa instruksi yang sama. Putaran tak terbatas biasanya digunakan dalam mikrokontroler dalam satu teks, yang ditulis sampai dengan kerangka bagian dari putaran tak terbatas. Dengan *BASIC Stamp2SX* dapat diselesaikan masalah putaran tak terbatas tanpa dihubungkan ke *power* atau program yang berbeda dijalankan.

### 3.4 Pengujian Motor Servo

Untuk melakukan pengujian motor servo ini, dilakukan dengan cara mengisikan program terlebih dahulu kepada mikrokontroler. Program tersebut adalah program untuk mengaktifkan motor servo. Seperti diketahui bahwa motor servo diaktifkan dengan memberikan sinyal PWM (*Pulse Width Modulation*) dengan lebar pulsa tertentu. Sinyal inilah yang akan memberitahukan motor servo harus berputar CW (Clock Wise = searah jarum jam) atau CCW (Counter Clock Wise = berlawanan arah jarum jam).

Putaran motor servo mempunyai 2 arah, yaitu searah dengan jarum jam dan berlawanan arah dengan jarum jam. Arah putaran motor servo ditentukan dengan cara uji coba-coba (*trial and error*) dari beberapa nilai *pulsout* (pulsa keluaran) dengan menggunakan program, sebagai berikut:

“Program menentukan arah putaran motor servo “

```
low 12          ' set P12 to out - low ( P12= pin 12)
loop :         ' label for goto loop to return to'
```

```

pulsout 12, 1700    ' send 1,7 ms pulses
pause 500          ' every 500 ms
goto loop          ' send program to " loop " label

```

Pertama kali menentukan *pulsout*, agar motor servo dalam keadaan diam, hal dikatakan bahwa motor servo dalam kondisi titik tengah arah putaran. Kemudian berikan nilai *pulsout* yang lebih besar atau kecil, sehingga terlihat arah putaran motor servo. Amati arah putaran motor servo searah atau berlawanan dengan jarum jam.

#### 4. Hasil dan Pembahasan

Motor servo kondisi diam, jika jumlah pulsa yang diberikan adalah 1600 pulsa (memberikan 1,6 ms pulsa). Untuk putaran motor servo searah dengan arah jarum jam, maka jumlah pulsa yang harus di-input-kan harus lebih kecil dari 1600 pulsa (1,6 ms pulsa). Sedangkan motor servo akan berputar berlawanan dengan arah jarum jam, maka jumlah pulsa yang di-input-kan harus lebih besar dari 1600 pulsa. Dari hasil eksperimen di atas dapat diaplikasikan untuk menggerakkan robot, adalah sebagai berikut:

- Gerakan maju, motor servo harus diberikan jumlah pulsa  $> 1600$  pulsa (1,6 ms pulsa)
- Gerakan mundur, motor servo harus diberikan jumlah pulsa  $< 1600$  pulsa (1,6 ms pulsa)

Hasil eksperimen yang dilakukan dengan meng-input-kan jumlah pulsa ke mikrokontroler didapatkan, sebagai berikut:

1. Hubungan antara jumlah pulsa yang di-input-kan dengan jarak yang ditempuh oleh motor servo bergerak maju.

Tabel 1. Hubungan jumlah pulsa dengan jarak tempuh

Jumlah pulsa servo	Periode	Jumlah Putaran	Waktu ( detik )	Kecepatan (cm/s)	Jarak yang ditempuh
1750	1000	15,25	23,5	12,25	287,41
1750	500	7,75	12	12,17	146,01
1750	100	1,50	2,4	11,78	18,84
1750	50	0,75	1,3	10,87	14,13

2. Hubungan antara jumlah pulsa yang di-input-kan dengan sudut putara motor servo

Tabel 2. Hubungan jumlah pulsa dengan jarak putaran motor

Jumlah pulsa servo	Periode	Sudut putaran servo (derajat)	Keterangan
1600	100	0	Diam
1800	50	30	Berputar berlawanan jarum jam
2000	50	60	Berputar berlawanan jarum jam
1700	100	90	Berputar berlawanan jarum jam
1700	100	180	Berputar berlawanan jarum jam
2000	100	270	Berputar berlawanan jarum jam
1600	100	0	Diam

Pulsa ini dapat dihasilkan dari pin OCR pada mikrokontroler. Perlu pengaturan register timer pada mikrokontroler agar dapat dihasilkan pulsa dengan lebar yang sesuai kita inginkan. Hal yang sangat penting adalah pengaturan frekuensi dan lebar pulsa on dan pulsa off. Oleh karena itu perlu dihitung berapa konstanta-konstanta timer yang di atur pada mikrokontroler. Dua parameter utama yang diperlukan untuk mencari konstanta-konstanta *timer* adalah nilai *clock* mikrokontroler dan nilai *clock timer*. Dari kedua parameter itu kita dapat merancang lebar pulsa *high* dan pulsa *low* dengan frekuensi tertentu yang sesuai untuk menggerakkan servo.

Pada tabel diatas merupakan tabel pengujian dengan menggunakan kontroler BASIC Stamp2SX dengan meng-input-kan jmlah pulsa untuk menggerakkan motor servo. Dengan memahami eksperimen ini, maka dapat membantu mahasiswa dalam mengembangkan sistem robotika. Kegiatan-kegiatan diatas tersebut di atas mendorong berkembangnya dunia per-robotan ditengah air terutama dikalangan mahasiswa maupun umum. Hal lain yang turut membantu mengembangkan dunia robot adalah terbitnya beberapa buku tentang robot itu sendiri.

## 5. Kesimpulan

Setelah melakukan tahap perencanaan dan pengujian motor servo, maka dapat diambil kesimpulan, bahwa: putaran motor servo yang dikendalikan mikrokontroler BASIC Stamp2SX sesuai dengan perintah yang dikehendaki. Untuk putaran motor servo searah dengan arah jarum jam, maka jumlah pulsa yang harus di-input-kan harus lebih kecil dari 1600 pulsa (1,6 ms pulsa). Sedangkan motor servo akan berputar berlawanan dengan arah jarum jam, maka jumlah pulsa yang di-input-kan harus lebih besar 1600 pulsa (1,6 ms pulsa).

## Daftar Pustaka

- [1] Anonymous, *Basic Stamp (Programming Manual Version 1.9)*, Parallax:inc.USA
- [2] Anonymous, *Robotic*.<http://parallaxin.com>
- [3] Anonymous, *Robotic*.<http://stampinclass>
- [4] Anonymous, <http://learncontrol.files.wordpress.com/2009/07/kendali-posisi-motor-dc.pdf>
- [5] Anonymous,[http://elektro\\_kontrol.blogspot.com/2011/06/realisasi\\_kontrol\\_pid\\_proporsional.html](http://elektro_kontrol.blogspot.com/2011/06/realisasi_kontrol_pid_proporsional.html)
- [6] Ogata, Katsuhiko. 1991. *Teknik Kontrol Automatik* (terjemahan: Ir. Edi Laksono), Jakarta: Erlangga
- [7] Gunterus, Frans.1994. *Falsafah Dasar: Sistem Pengendalian Proses*, Jakarta: PT. Elex Media Komputindo