bidang
REKAYASA

KARAKTERISTIK DAN PENGONTROLAN SERVOMOTOR

SYAHRUL
Jurusan Teknik Komputer
Universitas Komputer Indonesia

Servomechanism disingkat servo adalah suatu *device* yang digunakan untuk memberikan kontrol mekanik pada jarak. Servomotor mempunyai keluaran *shaft* (poros). Poros ini dapat ditempatkan pada posisi sudut spesifik dengan mengirimkan sinyal kode pada saluran kontrol servomotor. Selama sinyal kode ada di saluran kontrol, servo akan tetap berada di posisi sudut poros. Bila sinyal kode berubah, posisi sudut poros berubah. Aplikasi servo banyak ditemui pada *radio control* pesawatterbang model (*aeromodelling*), mobil *radio control*, boneka mainan, dan tentunya robot.

Servomotor ada yang DC, AC atau DC *brushless* yang dilengkapi dengan *device* pengindera posisi (yaitu *digital encoder*). Di sini akan dibahas tentang servomotor DC tiga-kawat. Servomotor DC tiga-kawat dibangun dari motor DC, *geartrain/gearbox*, *limit stop* yang melingkupi poros (tidak berputar), potensiometer sebagai umpanbalik (*feedback*) posisi, dan rangkaian terpadu untuk kontrol. Tiga kawat yang berada di luar yaitu untuk *power*, *ground* dan input kontrol untuk pemberian sinyal PWM (*pulse width modulation*)

Katakunci: servomotor, kontrol, pwm

PENDAHULUAN

Servomotor banyak digunakan dalam dunia robotika, karena selain ukurannya kecil, juga sangat tangguh. Servomotor standar seperti Futaba S-148 mempunyai torsi (*torque*) 42 oz/inch, yang merupakan servomotor yang sangat kuat untuk ukuran tersebut. Servomotor juga mengkonsumsi daya yang sebanding dengan beban mekanik. Dengan beban yang kecil, konsumsi daya tidak besar.

Salahsatu perbedaan utama antara servomotor dan steppermotor adalah bahwa servomotor, dari definisi, dijalankan dengan menggunakan *control loop* dan memerlukan sejumlah umpanbalik. *Control loop* menggunakan umpanbalik dari motor untuk membantu motor memperoleh keadaan (*state*) yang diinginkan (posisi, kecepatan dan sebagainya). Ada beberapa jenis *control loop*. Pada umumnya *control loop* PID (*proportional, Integral, Derivative*) digunakan untuk servomotor.

Bila menggunakan *control loop* misalnya PID, anda dapat melakukan *tune* pada servomotor. *Tuning* adalah proses dari pembuatan respon motor dalam cara yang diinginkan. Melakukan *tuning* pada motor dapat menjadi sangat sulit dan proses yang menjemukan, tetapi juga menguntungkan dalam hal memungkinkan user mempunyai kontrol yang lebih terhadap tingkahlaku motor.

Karena servomotor mempunyai *control loop* untuk memeriksa apakah *state* mereka berada di dalam, mereka

umumnya lebih andal daripada steppermotor. Bila steppermotor gagal dalam suatu step karena suatu hal, tidak ada *control loop* untuk mengkompensasi gerakan. *Control loop* pada servomotor secara tetap memeriksa apakah motor dalam lintasan yang benar dan jika tidak maka dilakukan *adjusment* yang diperlukan.

Pada umumnya, servomotor berjalan lebih halus daripada steppermotor kecuali digunakan *microstepping*. Lagi pula ketika kecepatan meningkat, torsi servomotor tetap konstan, membuat servomotor lebih baik daripada steppermotor pada kecepatan tinggi (biasanya di atas 1000 RPM, *rotary per minute*).

KEUNTUNGAN SERVOMOTOR

Beberapa keuntungan servomotor dibandingkan dengan steppermotor adalah:

- *High intermittent torque*
- Torsi tinggi untuk *inertia ratio*
- Kecepatannya tinggi
- Bekerja baik untuk kontrol kecepatan
- Tersedia dalam banyak ukuran
- Tidak bising

KEKURANGAN SERVOMOTOR

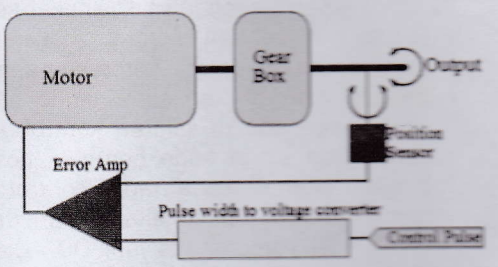
Beberapa kekurangan servomotor dibandingkan dengan steppermotor adalah:

- Lebih mahal daripada steppermotor
- Tidak dapat bekerja open loop - dibutuhkan umpanbalik
- Memerlukan penyesuaian parameter-parameter *control loop*
- Memerlukan pemeliharaan yang lebih karena adanya brush pada motor DC brush

BAGAIMANA SERVOMOTOR BEKERJA?

Servomotor terdiri dari beberapa bagian utama: motor dan gearbox, sensor posisi, *error amplifier* dan *motor driver* serta sirkuit yang mendekode posisi yang diminta. Gambar 1 menunjukkan diagram blok servomotor (*typical*).

Radio control receiver system (atau kontroler lainnya) membangkitkan suatu pulsa yang lebarnya berubah sekitar setiap 20ms. Pulsa ini lebarnya biasanya antara 1 dan 2 ms. Lebar pulsa digunakan oleh servo untuk menentukan posisi rotasi yang dikehendaki.



Gambar 1. Diagram blok servomotor

Pulsewidth to voltage converter

Pulsa kontrol diumpankan ke *pulsewidth to voltge controller*. Sirkuit ini mengisi muatan kapasitor pada kecepatan konstan selama pulsa dalam keadaan *high*. Bila pulsa menuju *low*, muatan pada kapasitor diumpankan ke output via *buffer amplifier* yang sesuai. Ini pada dasarnya menghasilkan suatu tegangan yang sesuai dengan lebar pulsa yang dikenakan.

Sirkuit ditala untuk menghasilkan tegangan pada perioda 1 ms sampai 2 ms. Tegangan output disangga dan juga tidak turun secara signifikan antara pulsa-pulsa kontrol sehingga lebar waktu antara pulsa-pulsa tidaklah kritis.

Position Sensor

Arus posisi rotasi poros (keluaran servomotor) dibaca oleh sebuah sensor. Sensor ini biasanya adalah sebuah potensiometer (*variable resistor*) yang menghasilkan tegangan yang sesuai dengan sudut mutlak poros.

Sensor posisi kemudian mengumpankan nilai arus ke *error amplifier* yang membandingkan arus posisi dengan posisi yang diperintahkan dari *pulsewidth to voltage converter.*

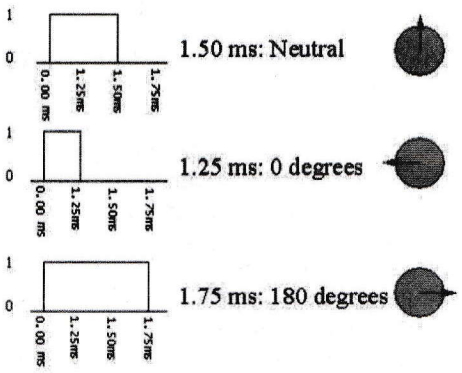
Potensiometer memungkinkan sirkuit kontrol untuk memonitor arus sudut servomotor. Jika poros motor berada pada sudut yang benar, maka motor mengunci. Jika sirkuit mendapatkan sudut yang tidak benar, dia akan memutar motor ke arah yang benar hingga sudutnya benar. Output poros servomotor dapat berputar sekitar 180 derajat. Umumnya, rentang putaran hingga 210 derajat tergantung pabrik. Servomotor normal digunakan untuk mengontrol gerakan sudut (*angular motion*) antara 0 dan 180 derajat.

Jumlah daya yang digunakan pada motor adalah sebanding dengan jarak yang ditempuh. Jadi, jika diperlukan poros berputar dengan jarak yang besar, motor akan berjalan pada kecepatan penuh. Jika diperlukan hanya putaran kecil, motor akan berjalan pada kecepatan lebih lambat. Ini disebut *proportional control.*

Saluran kontrol digunakan untuk berhubungan dengan sudut. Sudut ditentukan oleh durasi pulsa yang dikenakan pada saluran kontrol. Ini disebut *Pulse Width Modulation* (PWM). Servo mengharapkan mendapat pulsa setiap 20ms. Lebar pulsa menentukan seberapa jauh motor berputar. Sebagai contoh sebuah pulsa 1,5ms akan memutar motor ke posisi 90 derajat (sering disebut posisis netral). Jika pulsa lebih pendek dari 1,5ms maka motor akan memutar poros menuju 0 derajat. Jika pulsa lebih panjang dari 1,5ms maka poros berputar menuju 180 derajat.

Error Amplifier

Error amplifier adalah sebuah penguat operasional dengan umpanbalik negatif. Amplifier ini akan selalu mencoba memperkecil perbedaan antara input *inverting* dan *non-inverting* dengan men-*drive* outputnya ke arah yang benar.



Gambar 2.
Contoh diagram pewaktuan servomotor

Keluaran *error amplifier* digunakan untuk men-*drive* servomotor. Jika positip, servomotor akan berputar dalam suatu arah, jika negatip maka sebaliknya. Hal ini mengizinkan *error amplifier* mengurangi beda tegangan antara inputnya dan juga membuat servomotor menuju ke posisi yang diperintahkan.

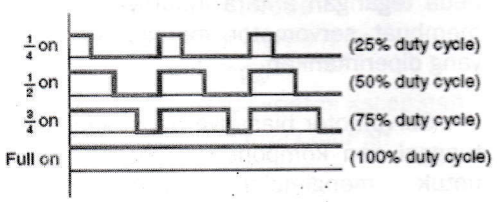
Servomotor biasanya berisi sebuah IC tunggal dan komponen-komponen diskrit untuk mengimplementasikan sistem kontrol keseluruhan.

Pengontrolan Servomotor dengan Mikrokontroler

Dari uraian di atas kita dapat menentukan bahwa kita perlu membangkitkan pulsa kira-kira setiap 20ms walaupun waktu aktual antara pulsa tidaklah kritis. Namun, lebar pulsa harus akurat untuk menjamin bahwa kita dapat secara akurat men-set posisi servomotor. Penggunaan mikrokontroler untuk keperluan pengotrolan servomotor dapat memudahkan di dalam desain dan implementasi pengontrolan.

Pulse Width Modulation

Pulse Width Modulation (PWM) merupakan pendekatan pengontrolan torsi dan kecepatan motor DC yang berbeda dengan teknik lainnya. Daya disuplai ke motor dalam bentuk sinyal gelombang persegi yang amplitudanya konstan tetapi lebar pulsanya atau *duty cycle* berubah-ubah. *Duty cycle* adalah persentasi waktu pulsa *high* terhadap perioda pulsa. Gambar 3 menunjukkan bentukgelombang untuk empat kecepatan berbeda. Untuk kecepatan yang paling rendah (lambat), daya disuplai pada seperempat waktu siklus (*duty cycle* 25%). Untuk *duty cycle* 50% (daya pada separuh waktu), motor akan berputar pada kecepatan setengah dan seterusnya. Pada praktisnya, faktor nonlinear menyebabkan motor berputar lebih lambat dari proporsi yang seharusnya.



Gambar 3. Bentukgelombang PWM

Banyak mikrokontroler yang dilengkapi dengan pembangkit PWM dan pada umumnya orang pada mulanya berfikir menggunakan ini untuk membangkitkan sinyal kontrol. Sayangnya dalam realitasnya tidak cocok.

Masalahnya adalah kita perlu pulsa pendek akurat yang relatif kemudian *delay* yang panjang, dan umumnya anda hanya mempunyai satu pembangkit PWM yang di -*share* antara beberapa servomotor yang akan memerlukan komponen-komponen *switching* di luar mikrokontroler dan membuat hardware menjadi kompleks.

Pembangkit PWM didesain untuk membangkitkan pulsa akurat dengan *duty cycle* antara 0% dan 100%, tetapi kita memerlukan dalam orde 5% sampai 10% (1ms/20ms sampai 2ms/20ms). Jika sebuah pembangkit PWM *typical* 8 atau 10 bit katakanlah demikian, maka kita hanya dapat menggunakan sebagian kecil bit ini untuk membangkitkan lebarpulsa yang kita perlukan dan juga kita akan kehilangan akurasi yang besar.

Timer

Pendekatan yang lebih menguntungkan dalam pengontrolan servomotor dapat diimplementasikan dengan timer sederhana dan interupsi perangkat lunak. Kuncinya adalah bahwa kita dapat menjalankan timer pada laju yang lebih cepat dan mengerjakan servomotor tunggal pada satu waktu, diikuti selanjutnya, dan seterusnya. Setiap output putarannya dikemudikan untuk waktu yang ditetapkan dan kemudian mati. Ketika semua output telah dikemudikan, maka siklus diulangi.

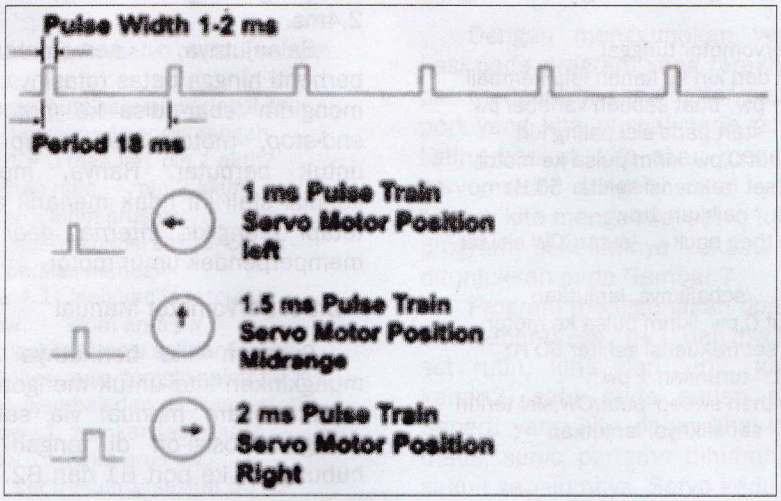
Pendekatan ini didemonstrasikan pada *PIC servo controller*.

Studi kasus: Menggunakan Servomotor dengan PIC Microcontroller

Servomotor pada dasarnya menggerakkan *gear motor dc* dengan kontrol umpanbalik posisi, yang memungkinkan untuk menempatkan rotor secara akurat, dengan rentang 90 derajat. Dapat juga dimodifikasi untuk rotasi kontinyu.

Servomotor mempunyai tiga kawat biasanya warna merah, hitam dan putih. Kawat merah untuk +VDC, hitam untuk *ground* dan putih untuk kontrol posisi. Sinyal kontrolnya adalah sinyal lebar-pulsa yang variable (*variable pulsewidth*) yang dapat diubah-ubah dari 1 hingga 2ms. Pulsa ini mengontrol posisi rotor.

Pulsa dengan lebar 1,0ms memutar poros dengan arah berlawanan jarum jam (*CCW-counterclockwise*). Pada pulsa 1,5ms menempatkan rotor pada posisi netral (0 derajat), dan pulsa 2ms akan menempatkan poros pada arah jarum jam



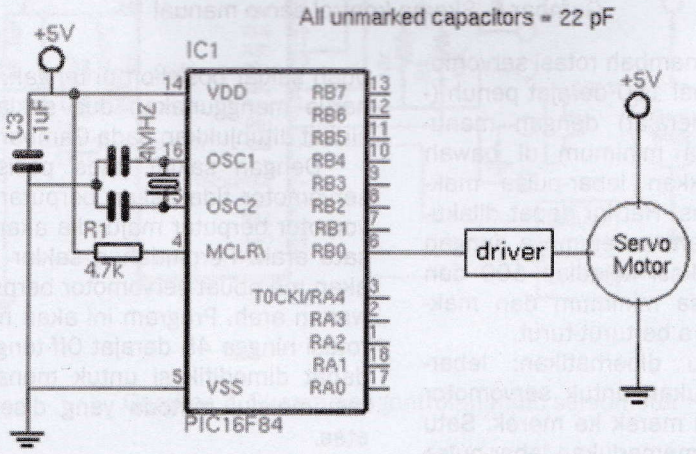
Gambar 4. Hubungan antara lebarpulsa dan posisi rotor

(*CW-clockwise*). Pulsa dikirim ke servomotor pada frekuensi sekitar 50 Hz. Hubungan antara lebar-pulsa dan posisi rotor dapat dilihat pada Gambar 4.

Pengontrolan Servomotor Tunggal

Pada studi kasus berikut kita akan melakukan *sweep* sederhana pada servomotor dari CCW ke CW dan *sweep* kembali. Program akan mendemonstrasikan secara sederhana prinsip pengontrolan servomotor dengan bahasa pemrograman PIC Basic (PIC Basic language). Skema sirkuit dapat dilihat pada Gambar 5.

Lebar-pulsa variabel (PW) mengontrol lebar-pulsa dan dimulai pada 100 (ekstrim kiri, -45 derajat). Program mengirimkan pulsa ke servomotor dan kemudian menaikkan nilainya 1 hingga mencapai 200 (ekstrim kanan, 45 derajat), pada titik dimana dia akan berputar balik.



Gambar 5. Antarmuka PIC16F84 dan servomotor

— Program Listing-1 —

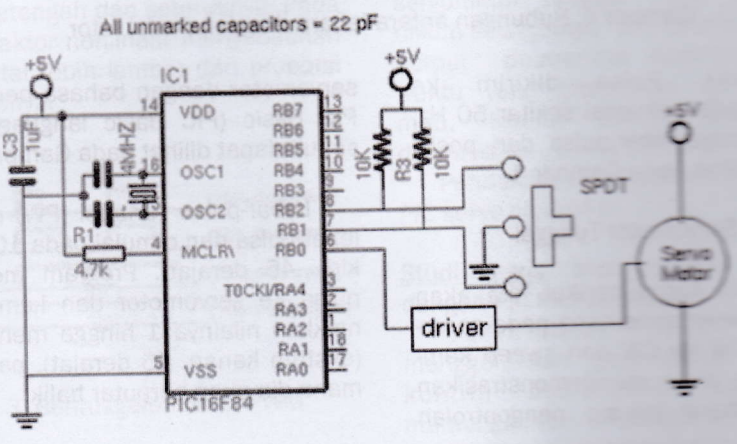
```
' Program servomotor tunggal
' Penyapuan dari kiri ke kanan lalu kembali
Symbol B1 = pw  'buat sebuah variabel pw
 pw = 100    'start pada sisi paling kiri
sweep: pulsout 0,pw 'kirim pulsa ke motor
 pause 18  'set frekuensi sekitar 50 Hz
 pw = pw + 1  ' naikkan 1 pw
 if pw > 200 then back     'putarCCW,sisi ter-
kanan
 goto sweep    'sebaliknya, lanjutkan
back: pulsout 0,pw  'kirim pulsa ke motor
 pause 18  'set frekuensi sekitar 50 Hz
 pw = pw – 1  ' turunkan 1 pw
 if pw < 100 then sweep 'putarCW,sisi terkiri
goto back    ' sebaliknya, lanjutkan
```

kan yang lainnya bisa hanya memerlukan 2,4ms.

Selanjutnya, servomotor akan berhenti hingga batas rotasinya. Jika anda mengirim lebar-pulsa ke motor melebihi *end-stop*, motor akan tetap mencoba untuk berputar. Hanya, motor pada kondisi stall ini tidak menarik arus lebih, tetapi mengikis *internal gear* sehingga memperpendek umur motor.

Kontrol Servomotor Manual

Program kita berikutnya akan memungkinkan kita untuk mengontrol servomotor secara manual via saklar SPDT (dengan posisi-off di tengah) yang dihubungkan ke port B1 dan B2. Tanpa sebuah saklar posisi-off di tengah, kita akan harus menggunakan dua saklar. Skema sirkuit ditunjukkan pada Gambar 6.



Gambar 6. Skema kontrol servo manual

Kita dapat menambah rotasi servomotor hingga mencapai 180 derajat penuh (-90 sampai 90 derajat) dengan menurunkan lebar-pulsa minimum di bawah 1ms dan menaikkan lebar-pulsa maksimum di atas 2ms. Hal ini dapat dilakukan dengan program sebelumnya dengan melakukan modifikasi kejadian 100 dan 200 ke lebar-pulsa minimum dan maksimum anda, secara berturut-turut.

Namun, perlu diperhatikan: lebar-pulsa yang diperlukan untuk servomotor berubah-ubah dari merek ke merek. Satu servomotor dapat memerlukan lebar-pulsa 2,8ms untuk rotasi maksimum, sedang-

Dengan saklar pada posisi tengah, servomotor tidak akan berputar. Bila servomotor berputar maju, dia akan berputar satu arah. Perpindahan saklar ke bawah akan membuat servomotor berputar berlawanan arah. Program ini akan membatasi rotasi hingga 45 derajat Off-tengah, tetapi dapat dimodifikasi untuk menambah rotasi melalui metoda yang disebutkan di atas.

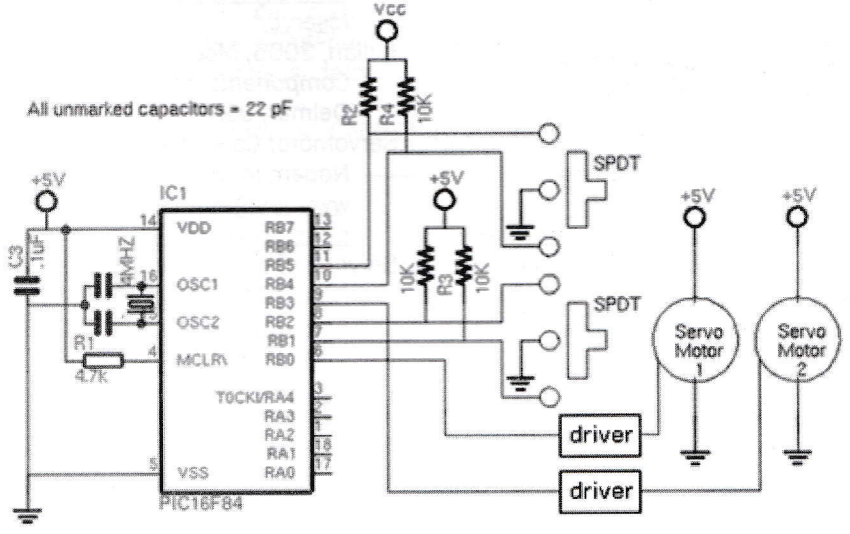
— Program Listing-2 —

```
' Kontrol Manual arah servo melalui sebuah
SPDT switch.
Symbol B1=pw 'buat sebuah variabel pw
pw = 150    'mulai pada posisi tengah
check: if pin1 = 0 then left 'pin 1 aktif?
 if pin2 = 0 then right     'pin 2 aktif?
 Pulsout 0,pw    'kirim arus pw
 pause 18    'set frekuensi sekitar 50 Hz
 goto che    'periksa kembali
left: pw = pw + 1   'naikkan lebarpulsa
 pulsout 0,pw       'kirim arus pw
 pause 18    'set frekuensi sekitar 50 Hz
 if pw > 200 then max  'tunggu sekitar 2 ms
 goto check    'kembali dan periksa lagi
right: pw = pw - 1  'trunkan lebarpulsa
 pulsout 0,pw       'kirim arus pw
 pause 18    'set frekuensi sekitar 50 Hz
 if pw < 100 then min 'tunggu di bawah 1 ms
 goto check          'periksa kembali
max: pw = 200          'buat pw pada 2 ms
 goto check          'periksa kembali
min: pw = 100          'buat pw pada 1 ms
 goto check          'periksa kembali
```

Pengontrolan Beberapa Servomotor

Dengan menggunakan versi modifikasi pada program yang terakhir, kita dapat mengontrol servomotor sebanyak I/O port yang kita punyai pada port B. Pada listing berikut, kita akan mengontrol dua servomotor dengan cara yang sama seperti kita mengontrol servo tunggal pada program sebelumnya. Skema sirkuitnya ditunjukkan pada Gambar 7.

Program menggunakan dua buah variabel lebar-pulsa, pw1 dan pw2, dan dua set rutin, kiri1 dan kiri2, kanan1 dan kanan2, satu pada setiap servomotor. Seperti yang kita dapat lihat dalam skematik, servo pertama dihubungkan pada sirkuit sebelumnya. Servo kedua sekarang menggunakan B3 sebagai pulsa keluar, B4 dan B5 untuk saklar SPDT.



Gambar 7. Skema pengontrolan multi servomotor

— Program Listing-3 —

```
'Kontrol Manual dua servomotor menggunakan 2
SPDT switch
'Gunakan B1 untuk menahan variabel lebar
pulsa untuk servo 1
'Gunakan B2 untuk menahan variabel lebar
pulsa untuk servo 2
'Inisialisasi Variabel
B1 = 150  'start servo 1 pd posisi tengah
B2 = 150  'start servo 2 pd posisi tengah
start:   'cek pada switch tertutup
 IF pin1 = 0 Then left1  'sw1 aktif kiri?
 IF pin2 = 0 Then right1 'sw1 aktif kanan?
 IF pin4 = 0 Then left2  'sw2 aktif kiri?
 IF pin5 = 0 Then right2 'sw2 aktif kanan?
 PulsOut 0, B1 'kirim arus servo 1
 PulsOut 3, B2 'kirim arus servo 2
 Pause 18
 GoTo start
'Rutin untuk Servomotor 1
left1:
 B1 = B1 + 1  'naikkan lebarpulsa
 PulsOut 0, B1 'kirim arus B1
 PulsOut 3, B2 'kirim arus B2
 Pause 18  'set frekuensi update 50 hz
 IF B1 > 225 Then max1 'max 2.25 ms
 GoTo start
right1:
 B1 = B1 - 1  'turunkan lebarpulsa
 PulsOut 0, B1 'kirim arus B1
 PulsOut 3, B2 'kirim arus B2
 Pause 18  'set frekuensi update 50 hz
 IF B1 < 75 Then min1 'min 0,75 ms
 GoTo start
max1:
 B1 = 225  'cap max B1 pada 2.25 ms
 GoTo start
min1:
 B1 = 75   'cap min B1 pada 0.75 ms
 GoTo start
'Rutin untuk Servomotor 2
left2:
 B2 = B2 + 1   'naikkan lebarpulsa
 PulsOut 0, B1  'kirim arus B1
 PulsOut 3, B2  'kirim arus B2
 Pause 18  'set frekuensi update 50 hz
 IF B2 > 225 Then max2   'max 2.25 ms
 GoTo start
right2:
 B2 = B2 - 1   'turunkan lebarpulsa
 PulsOut 0, B1  'kirim arus B1
 PulsOut 3, B2  'kirim arus B2
 Pause 18  'set frekuensi update 50 hz
 IF B2 < 75 Then min2    'min 0,75 ms
 GoTo start
max2:
 B2 = 225      'cap max B1 pada 2.25 ms
 GoTo start
min2:
 B2 = 75       'cap min B1 pada 0.75 ms
 GoTo start
```

PENUTUP

Dengan pengontrolan servomotor menggunakan mikrokontroler dapat diperoleh kemudahan dalam desain dan implementasi pengontrolan, yaitu dari segi kepresisian/efektivitas tundaan waktu (*delay*) menggunakan PWM yang dapat dibangkitkan melalui timer. Kecepatan servomotor dapat diatur dengan mudah berdasarkan perubahan lebarpulsa yang dapat diatur menggunakan mikrokontroler. Di samping itu memberikan efisiensi sirkuit karena mengurangi penggunakan sirkuit eksternal.

DAFTAR PUSTAKA

How R.C. servo motors work, diakses tanggal 19 Nopember 2006, http://www.digitalnemesis.com/info/docs/rcservo/

Killian, 2005, *Modern Control Technology: Components and Systems*, 2nd edition, Delmar, USA.

Servomotor Controlling, diakses tanggal 19 Nopember 2006, http://www.imagesco.com/articles/picservo/01.html

Servomechanism, diakses tanggal 19 Nopember 2006, http://en.wikipedia.org/wiki/Servomechanism

Servomotor overview, diakses tanggal 19 Nopember 2006, http://zone.ni.com/devzone/cda/ph/p/id/233

Servo motors, diakses tanggal 19 Nopember 2006, http://www.electricmotors.machinedesign.com/guiEdits/Content/bdeee4a/bdeee4a_1.aspx

Whats a servo?, diakses tanggal 19 Nopember 2006, http://www.seattlerobotics.org/guide/servos.html.