

Pengendali Motor Servo Posisi dengan Kendali Pid Berbasis Mikrokontroler Atmega

Rahmat^{1*}, Wiyono²

^{1,2} Program Studi Teknik Elektronika, Akademi Teknologi Warga Surakarta

*Email: rahmatteatw@gmail.com

Abstrak

Keywords:
Motor servo;
Mikrokontroler
AVR; Pengendalian
posisi

Proses pengendalian gerakan dalam banyak penerapan sering menggunakan motor sebagai penggerak. Pengendalian gerakan yang memerlukan keakuratan posisi yang diinginkan dalam banyak penerapan menggunakan motor servo. Kelebihan penggunaan motor servo sebagai penggerak dapat dikendalikan pada posisi yang akurat. Penerapan konsep kendali yang dipelajari pada teori sistem kendali yang dijabarkan dalam bentuk pernyataan matematik salah satunya algoritma Proportional Integral dan Derivatif yang dikenal dengan kendali PID yang banyak digunakan dalam berbagai aplikasi proses kendali. Dalam penelitian ini dibuat prototype modul kendali PID sebagai perangkat kendali posisi motor servo sebagai gambaran penerapan nyata dari konsep kendali PID yang diterapkan dalam pengendalian posisi motor servo. Mikrokontroler sebagai perangkat pusat kendalinya menggunakan mikrokontroler AVR Atmega. Penelitian ini menghasilkan prototype penerapan algoritma kendali PID sebagai konsep kendali yang direalisasikan dalam bentuk rangkaian yang nyata dalam bentuk rangkaian berbasis mikrokontroler dengan perangkat yang dikendalikan motor servo untuk mengatur posisi / keadaan motor servo sesuai dengan yang diinginkan. Pada penelitian dan pengujian ini masih merupakan prototype untuk merealisasikan konsep pengendalian dengan kendali PID. Dari hasil penelitian ini telah membuktikan konsep kendali PID untuk diterapkan pada kendali motor servo untuk mencapai posisi yang diinginkan dengan kecepatan respon dengan mengubah nilai parameter K_p , K_i dan K_d . Dari hasil perancangan dan uji coba kendali motor servo dengan kendali mikrokontroler dapat disimpulkan kecepatan mencapai keadaan stabil (Steady State) tergantung pada nilai parameter K_p , K_i , dan K_d . Semakin besar nilai K_p akan menambah kecepatan mencapai steady state (nilai kestabilan yang diinginkan). Semakin besar nilai K_d akan mengurangi harga penyimpangan dari nilai steady state. Penentuan kombinasi K_p , K_i dan K_d akan menghasilkan kecepatan mencapai nilai steady state dengan nilai simpangan terkecil.

1. PENDAHULUAN

Dalam proses pengendalian gerakan dalam banyak penerapan sering menggunakan motor sebagai penggerak. Pengendalian gerakan yang memerlukan keakuratan posisi yang diperlukan/diinginkan dalam banyak penerapan menggunakan motor servo. Kelebihan penggunaan motor servo sebagai penggerak dapat dikendalikan pada posisi yang diinginkan pada posisi yang akurat.

Proses pengendalian motor sebagai objek yang akan dikendalikan dalam penerapan kendali yang kompleks banyak menggunakan Mikrokontroler sebagai perangkat terprogram

yang bisa menerapkan algoritma kendali dalam bentuk rangkaian yang nyata. Penerapan konsep kendali yang dipelajari pada teori sistem kendali yang dijabarkan dalam bentuk pernyataan matematik salah satunya algoritma Proportional, Integral dan Derivatif yang dikenal dengan kendali PID yang banyak digunakan dalam berbagai aplikasi proses kendali. Algoritma kendali PID dalam aplikasi pengendalian motor lebih khusus kendali motor servo, algoritma PID dibuat dalam program yang diisikan dalam rangkaian mikrokontroler yang akan difungsikan sebagai pusat proses rangkaian kendali.

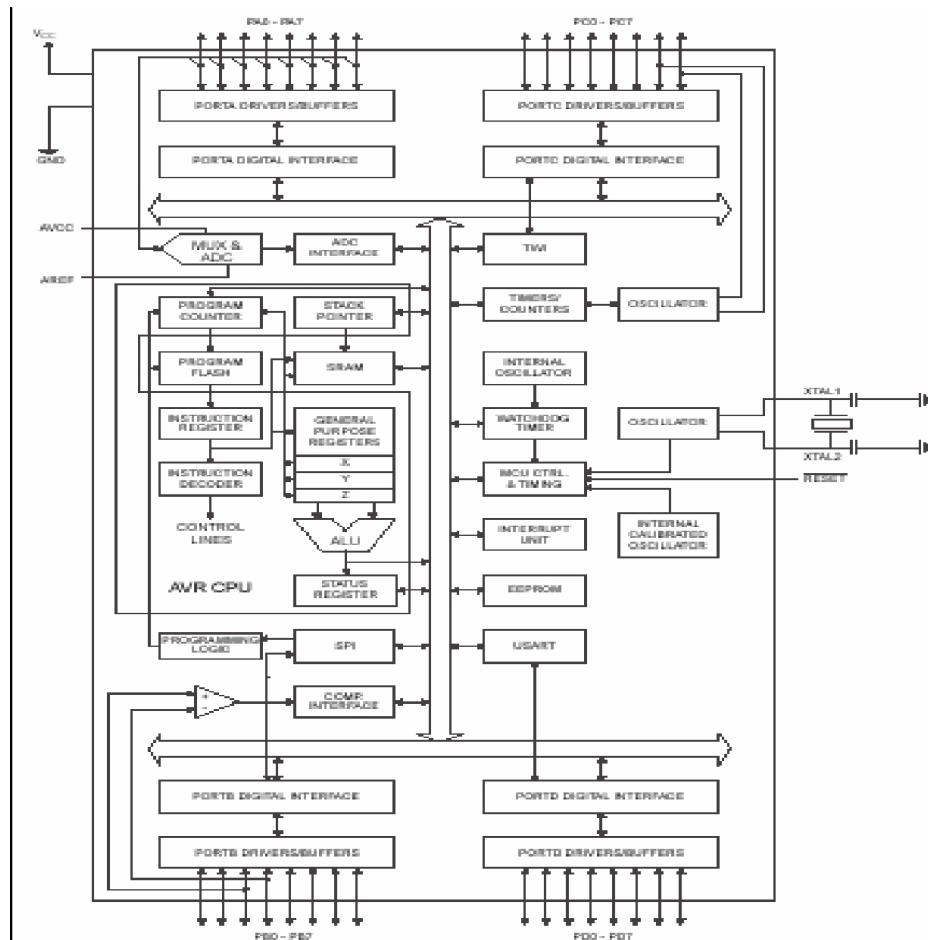
Dalam penelitian ini akan dibuat prototype modul kendali PID sebagai perangkat kendali posisi motor servo sebagai gambaran penerapan nyata dari konsep kendali PID yang diterapkan dalam pengendalian posisi motor servo. Mikrokontroler sebagai perangkat pusat kendalinya menggunakan mikrokontroler AVR Atmega. Dari uraian singkat diatas maka penelitian ini akan menghasilkan prototype penerapan algoritma kendali PID sebagai konsep kendali yang direalisasikan dalam bentuk rangkaian yang nyata dalam bentuk rangkaian berbasis mikrokontroler dengan perangkat yang dikendalikan motor servo untuk mengatur posisi / keadaan motor servo sesuai dengan yang diinginkan.

ATMEL AVR ATmega32

Mikrokontroler ATmega32 merupakan mikrokontroler generasi AVR (*Alf and Vegard's Risk processor*). Mikrokontroler AVR memiliki arsitektur RISC (*Reduced Instruction Set Computing*) 8 bit, semua instruksi dikemas dalam kode 16-bit (*16-bits word*) dan sebagian besar instruksi dieksekusi dalam 1 siklus clock. Pada perancangan sistem kontrol ini dipilih Mikrokontroler jenis ATMEL AVR RISC dengan pertimbangan sebagai berikut:

1. ATMEL AVR RISC memiliki fasilitas lengkap dan harga relatif murah.
2. Kecepatan maksimum eksekusi instruksi mikrokontroler mencapai 16 MIPS (*Million Instruction per Second*),
3. Konsumsi daya yang rendah jika dibandingkan dengan kecepatan eksekusi instruksi.
4. Ketersediaan kompilator C (CV AVR) yang memudahkan user untuk memprogram menggunakan bahasa C.

ATmega32 memiliki 4 buah port input/output 8 bit, yaitu PORTA, PORTB, PORTC, dan PORTD. Selain sebagai input/output masing masing port juga memiliki fungsi yang lain. PORTA dapat difungsikan sebagai ADC (*Analog to Digital Converter*), PORTB dapat difungsikan sebagai SPI (*Serial Peripheral Interface*) communication. Fungsi-fungsi yang lain dapat dilihat pada *datasheet* ATmega32.



Gambar 1. Diagram Blok Fungsional ATmega32

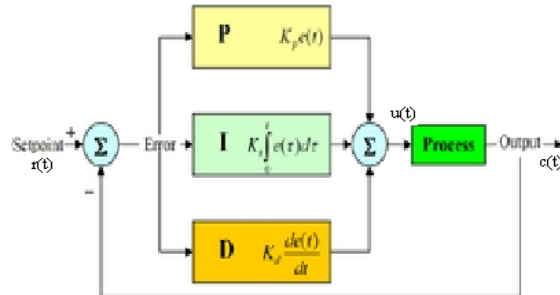
Pada Gambar 1 memperlihatkan ATmega32 memiliki bagian sebagai Berikut:

1. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 buah register.
5. *Watchdog Timer* dengan osilator internal.
6. SRAM sebesar 2 Kbyte.
7. Memori *Flash* sebesar 32 Kb dengan kemampuan *Read Write*.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI.
10. EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1K byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial, kecepatan maksimal 2,5 Mbps.
13. Mikroprocessor 8 bit berbasis RISC kecepatan maksimal 16 MHz.

Perangkat input/output digunakan oleh prosessor untuk berkomunikasi dengan dunia luar. Fasilitas input/output merupakan fungsi Mikrokontroler untuk dapat menerima sinyal masukan (input) dan memberikan sinyal keluaran (output). Contoh pemakaian perangkat I/O adalah serial komunikasi pada keyboard, tampilan LCD dan sinyal interupsi dari luar.

Pengendali PID

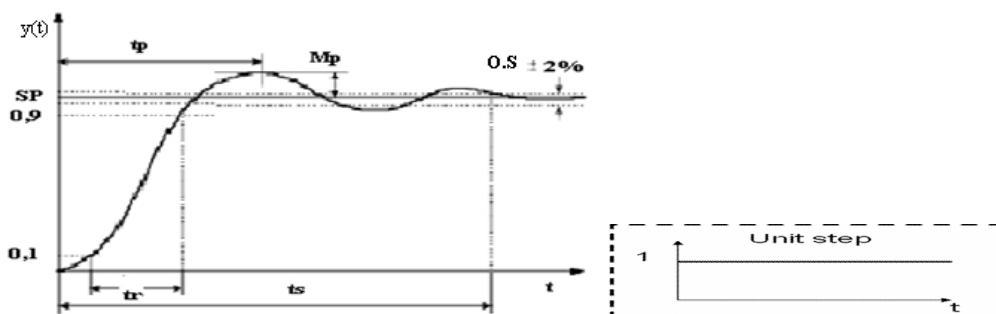
Pengendali PID merupakan jenis pengendali yang bekerja secara kontinu. Sesuai dengan cara sistem kendali menghasilkan aksi kendali, ada tiga jenis pengendali kontinu, yaitu pengendali proporsional disingkat P, pengendali integral disingkat I dan pengendali diferensial disingkat D.



Gambar 2. Diagram kotak PID analog

Dari analisis respon sistem yang telah dilakukan, bagaimana respon sistem ($c(t)$) yang diinginkan apakah sesuai dengan input ($r(t)$) (misal : unit step) seperti diagram kotak Gambar 2, jika tidak sesuai maka salah satu cara adalah menambahkan kendali. Fungsi kendali adalah mengendalikan sistem dengan memanipulasi sinyal *error*, sehingga respon sistem (*output*) sama dengan yang diinginkan (*input*). Kendali merupakan “Otak” dari sistem yang menerima *error* ($e(t)$) sebagai input kemudian menghasilkan sinyal kendali ($u(t)$). $U(t)$ menyebabkan *controlled variable* ($c(t)$) menjadi sama dengan *set point* ($r(t)$)

Dari analisis respon sistem, kinerja dari sistem kendali mampu mengatasi masalah Kestabilan, *Respon transient* (karakteristik sistem) dan *Error steady state*. Analisis dimaksudkan untuk menguji kelayakan sistem kendali dilihat berdasarkan karakteristik respon sistem kendali Gambar dibawah ini.



Gambar 3. Karakteristik respon sistem kendali

Waktu kenaikan (t_r): waktu saat *controlled variable* pertama kali mencapai *set point*.

Waktu puncak (t_p): waktu saat *controlled variable* mencapai lewatan maksimum.

Waktu penetapan (t_s): waktu saat *controlled variable* menetap (stabil) dalam daerah sekitar harga akhir yang ditentukan (2% dari set posisi).

Lewatan Maksimum (M_p) : Posisi tertinggi yang dicapai *controlled variable* diukur dari set posisi.

Output Steady state Error (O.S) : Kesalahan saat *controlled variable* menetap (stabil).

Misalkan sistem kendali diberi tanggapan *unit step* maka respon yang diinginkan diharapkan stabil, stabil artinya tanggapan sistem tidak berosilasi, atau cenderung semakin menjauhi *set point*. Sedangkan Karakteristik respon transient diharapkan:

M_p : 0 % (sekecil mungkin), T_r , t_p , t_s : 0 (sekecil mungkin), *steady state error*: 0 (tidak ada *steady state error*)

Kendali Proporsional

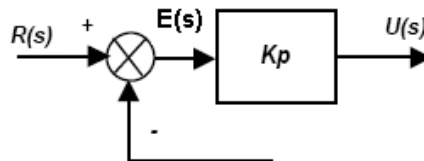
Istilah proporsional dikaitkan dengan perubahan aksi kendali yang proporsional dengan nilai *error* saat ini. Respon proporsional dapat diatur dengan mengalikan *error* dengan suatu konstanta K_p , disebut dengan penguatan proporsional. Aksi kendali proporsional dinyatakan dengan,

$$U(t) = K_p e(t)$$

Atau dalam bentuk fungsi alih (*Transfer Function* : "TF")

$$U(s)/E(s) = K_p$$

$E(s)$: *Error signal*; $U(s)$: *Controller signal*; K_p : *Gain Over of Proportional*.



Gambar 4. Diagram kotak kendali Proporsional

Penguatan proporsional yang tinggi menghasilkan perubahan aksi kendali yang tinggi untuk tiap-tiap perubahan *error* yang terjadi. Jika penguatan proporsional terlalu besar dapat menyebabkan sistem menjadi tidak stabil. Kebalikannya, penguatan kecil akan menghasilkan respon keluaran yang kecil terhadap perubahan *error* yang besar, dan mengurangi sensitivitas pengendali.

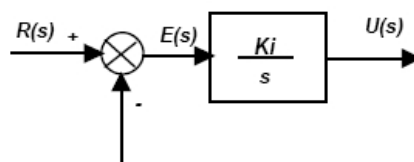
Kendali Integral

Kendali integral memberikan kontribusi aksi kendali proporsional dengan nilai *error*. Penjumlahan *error* setiap saat memberikan akumulasi *offset* yang dikoreksi sebelumnya. Akumulasi *error* dikalikan dengan penguatan integral dan ditambahkan ke keluaran pengendali. Besarnya kontribusi pengendali integral terhadap aksi kendali secara keseluruhan ditentukan oleh penguatan integral, K_i . Pengendali integral dinyatakan dengan:

$$U(t) = K_i \int_0^t e(\tau) d\tau$$

Dalam bentuk fungsi alih

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$



Gambar 5. Diagram kotak kendali Integral

Pengendali integral yang ditambahkan ke pengendali proporsional mempercepat pergerakan proses menuju *set point* dan menghilangkan sisa *steady state error* yang muncul pada pengendali proporsional murni.

Dalam fungsi matematik kendali tipe PI dapat dinyatakan:

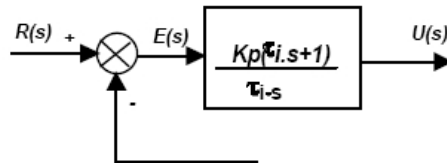
$$U(t) = K_p [e(t) + \frac{1}{\tau_i} \int e(t) dt]$$

Dalam fungsi alih dapat dinyatakan:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s}\right) = \frac{K_p (\tau_i s + 1)}{\tau_i s}$$

τ_i : merupakan konstanta waktu untuk kendali integral dan K_p adalah penguatan proporsional.

Secara diagram kotak dapat digambarkan sebagai berikut :



Gambar 6. Diagram kotak kendali Proporsional Integral

Kendali Derivatif

Kendali derivatif memberikan kontribusi aksi kendali proporsional dengan nilai durasi *error*. Laju perubahan proses *error* dikalkulasikan dengan menentukan slope terhadap waktu dan mengalikannya dengan penguatan derivatif K_d .

$$U(t) = K_d \frac{de}{dt}$$

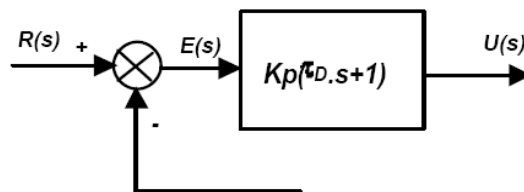
Dalam praktek, kendali derivatif sering digabungkan dengan kendali proporsional. sehingga secara matematik dapat dinyatakan :

$$U(t) = K_p \cdot e(t) + \tau_D \frac{d}{dt} e(t)$$

Dalam fungsi alih kendali PD dapat dinyatakan:

$$\frac{U(s)}{E(s)} = K_p (\tau_D \cdot s + 1)$$

Sehingga diagram kotak dapat digambarkan sebagai berikut :



Gambar 7. Diagram kotak kendali Proporsional Derivatif

Besarnya kontribusi pengendali derivatif terhadap aksi kendali keseluruhan disebut dengan penguatan derivatif K_d .

Keluaran tiga jenis pengendali ini dijumlahkan untuk mendapatkan keluaran pengendali PID. Dengan mendefenisikan $u(t)$ sebagai keluaran pengendali diperoleh bentuk akhir keluaran pengendali PID sebagai berikut:

$$U(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}$$

dan parameter yang dapat ditala adalah:

K_p : Penguatan proporsional, K_p yang lebih besar biasanya menyebabkan respon yang lebih cepat pada saat *error* yang besar. Penguatan proporsional yang terlalu besar menyebabkan proses tidak stabil dan berosilasi. K_i : Penguatan integral, K_i yang lebih besar menyebabkan *steady state errors* dihilangkan lebih cepat. K_d : Penguatan derivatif yang lebih besar K_d

mengurangi *overshoot*, tetapi melambatkan respon transien dan dapat menyebabkan ketidakstabilan yang disebabkan noise.

Jika pemilihan parameter pengendali tidak tepat, maka dapat menyebabkan sistem menjadi tidak stabil. Penalaan parameter kendali ini menentukan respon kendali yang dihasilkan. Pemilihan metode penalaan dapat dilakukan berdasarkan sistem yang akan dikendalikan.

Motor Servo

Motor servo adalah sebuah motor dengan sistem closed feedback di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, 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.

Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinu seperti motor DC maupun motor stepper. Walau demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinu. Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagian-bagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan duty cycle sinyal PWM pada bagian pin kontrolnya.

2. METODE

Langkah Penelitian

Untuk merealisasikan penelitian dilakukan langkah penelitian sebagai berikut:

1. Pembuatan program yang akan diisikan / didownload pada rangkaian minimum sistem Mikrokontroler sebagai rangkaian kendali motor servo.
2. Mengecek program assembler sampai tidak ada kesalahan.
3. Mengisi program ke Mikrokontroler.
4. Uji coba karakteristik sinyal output dari modul mikrokontroler sebagai sinyal kendali motor servo.
5. Uji coba pengendalian motor servo dengan modul AVR untuk berbagai posisi yang diinginkan, pengambilan data percobaan.

Alat dan bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah:

1. Osiloskop.
2. Personal Komputer.
3. Downloader.
4. Motor servo DC sebagai perangkat yang akan dikendalikan kontroler.
5. Rangkaian Mikrokontroler dengan komponen utama mikrokontroler AVR yang sudah terisi program untuk mengendalikan motor servo.

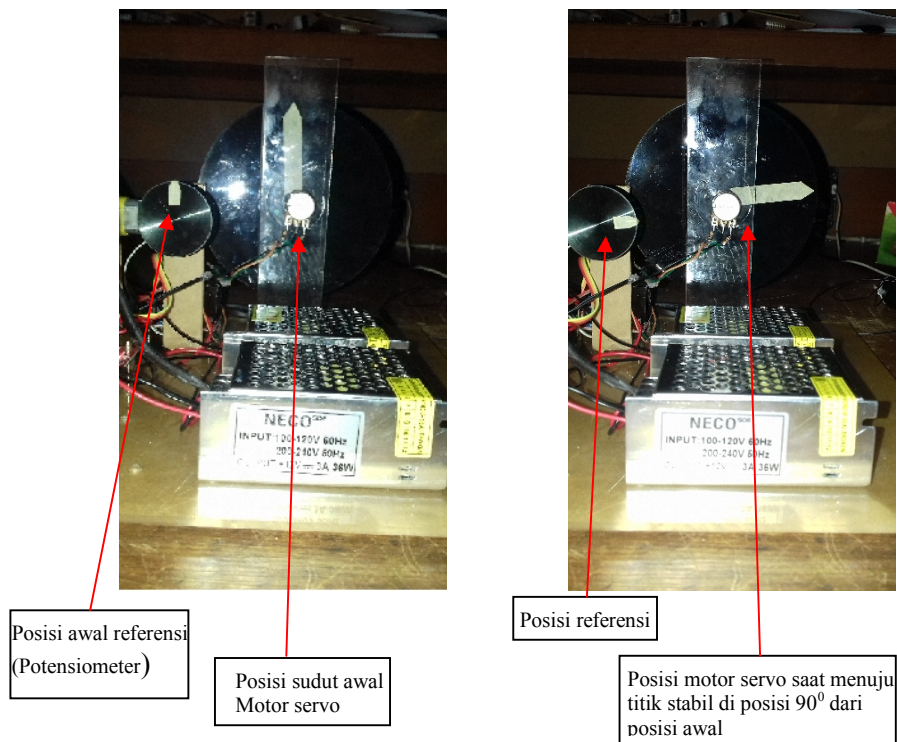
3. HASIL DAN PEMBAHASAN

Perancangan dalam penelitian ini adalah membuat modul Mikrokontroler AVR sebagai rangkaian kendali posisi motor servo DC. Perancangan dan pembuatan modul kendali dengan merakit minimum sistem AVR, driver motor servo, motor servo dengan gear rasionya. Adapun modul Kendali Mikrokontroler yang dibuat seperti ditunjukkan pada gambar dibawah ini



Gambar 8. Prototype rangkaian pengendali motor servo posisi

Data hasil uji coba modul mikrokontroler yang sudah dirakit untuk mengendalikan posisi motor servo (berputar pada sudut yang diinginkan) telah dilakukan dengan menginputkan data parameter kendali K_p , K_i , dan K_d pada modul Mikrokontroler sebagai controller PID. Dari beberapa sampel data pengujian untuk seting parameter PID ditunjukkan pada gambar dibawah ini.

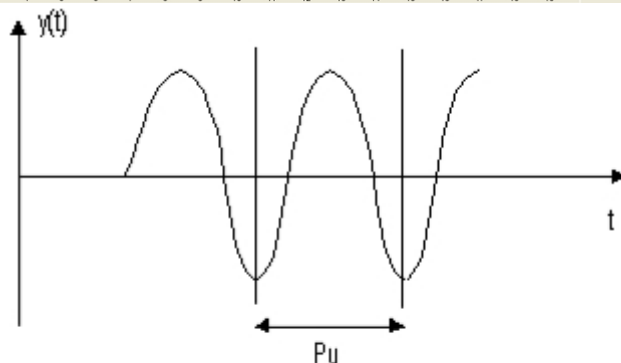


Gambar 9. Uji coba pengaturan posisi motor servo dari sudut 90° ke 180°



Gambar 10. Uji coba penelusuran posisi dari sudut 0° menuju 180°

Berikutnya analisa data hasil pengujian untuk plotting kurva pengendalian dari posisi awal sampai pada posisi motor servo yang diinginkan dengan dengan merubah nilai parameter K_p , K_i dan K_d . Sesuai metoda Ziegler –Nichols dengan metoda osilasi didapatkan sampling data dengan mengatur nilai K_p supaya terjadi sustain oscillation didapat nilai $K_u = 10$ hasil uji ditunjukkan pada gambar dibawah ini



Kurva respon sustain oscillation

Gambar 11. Kurva Osilasi untuk mencari nilai P_u

Penalaan parameter PID didasarkan terhadap kedua konstanta hasil eksperimen K_u dan P_u . Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p , T_i dan T_d berdasarkan rumus yang ditunjukkan pada tabel 1

Tabel1. Penalaan parameter PID dengan metoda isolasi

Tipe KONTROLLER	K_p	T_i	T_d
P	$0.5 * K_u$		
PI	$0.45 * K_u$	$0,5 * P_u$	
PID	$0,6 * K_u$	$0,5 * P_u$	$0,125 * P_u$

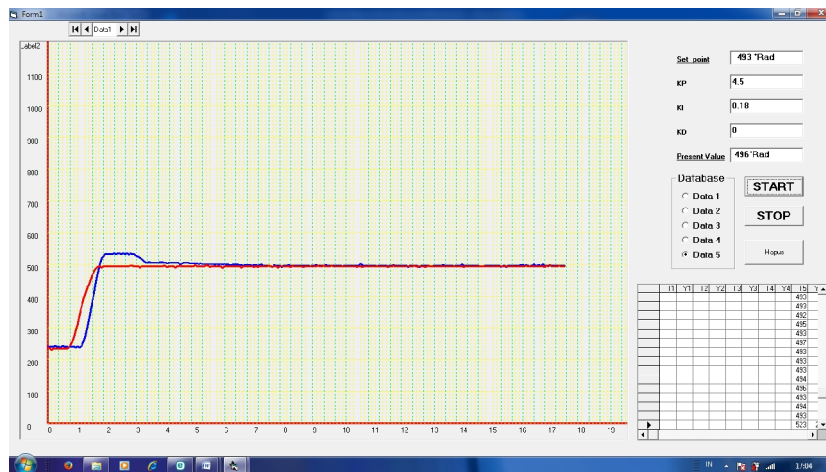
Dari hasil uji dengan memasukan parameter $K_u=10$ kedalam tabel dengan tipe kontroller PI (Proporsional Integral) dapat dihitung sebagai berikut

$$K_p = 0,45 * K_u = 0,45 * 10 = 4,5$$

$$T_i = 0,5 * P_u = 0,5 * 0,5 = 0,25$$

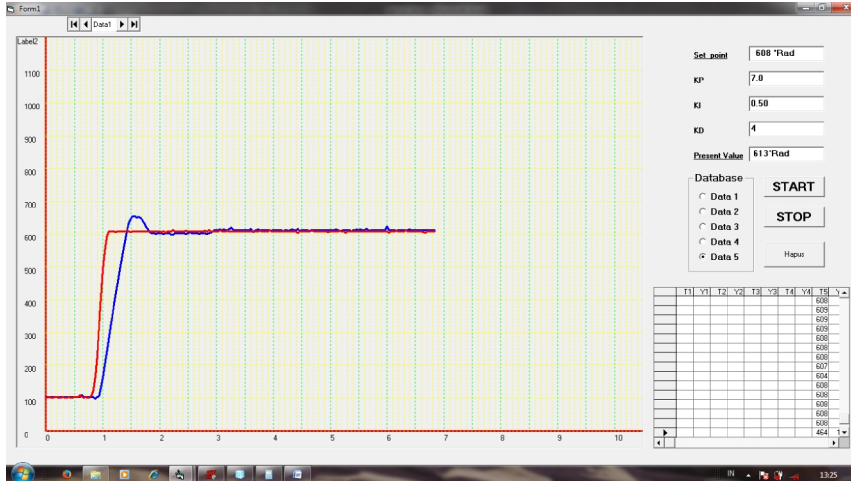
$$K_i = K_p / T_i = 4,5 / 0,25 = 18$$

Dari seting sesuai tabel diatas didapatkan hasil pengujian seperti gambar dibawah ini

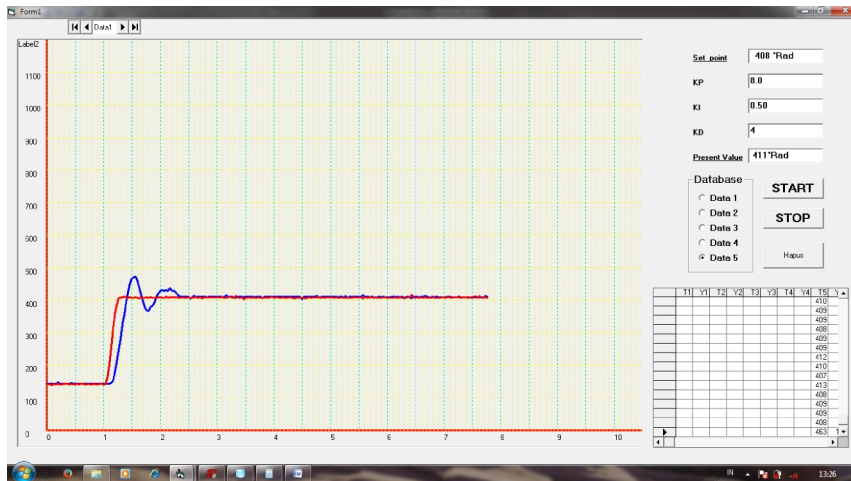


Gambar 12. Kurva pengendalian posisi motor servo dengan parameter $K_p = 4.5$ $K_i = 0.18$ $K_d = 0$

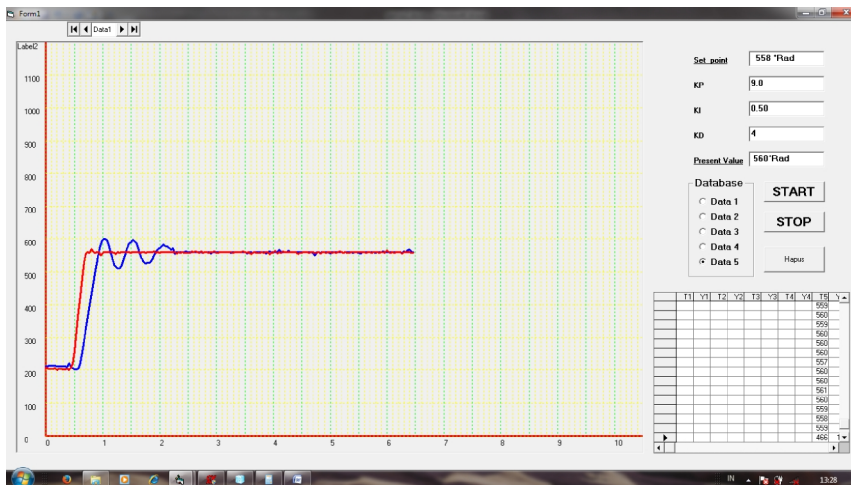
Dari data yang tertampil nilai steady state eror yang relatif besar 5 dt. Selanjutnya penulis melakukan pengujian dengan variasi nilai parameter PID seperti tertampil pada gambar dibawah.



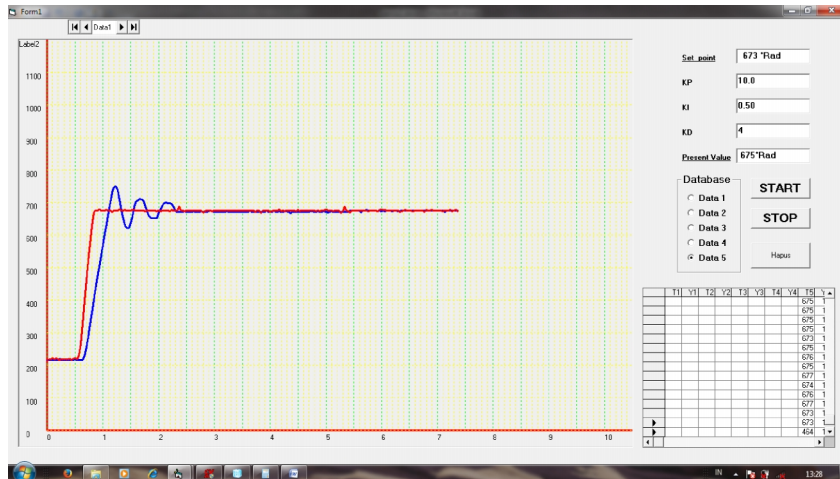
Gambar 13. Kurva pengendalian posisi motor servo dengan parameter $K_p = 7$ $K_i = 0.5$ $K_d = 4$



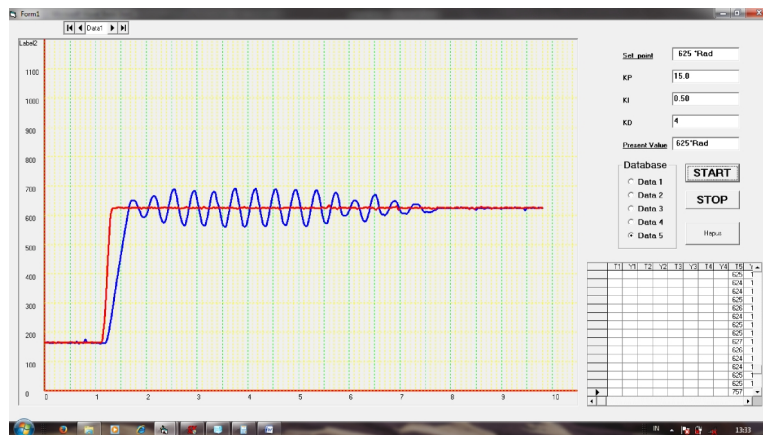
Gambar 14. Kurva pengendalian posisi motor servo dengan parameter $K_p = 8$ $K_i = 0.5$ $K_d = 4$



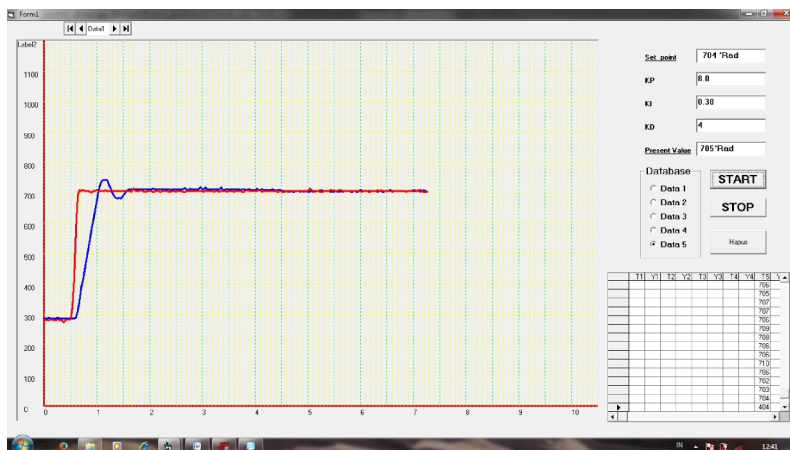
Gambar 15. Kurva pengendalian posisi motor servo dengan parameter $K_p = 9$ $K_i = 0.5$ $K_d = 4$



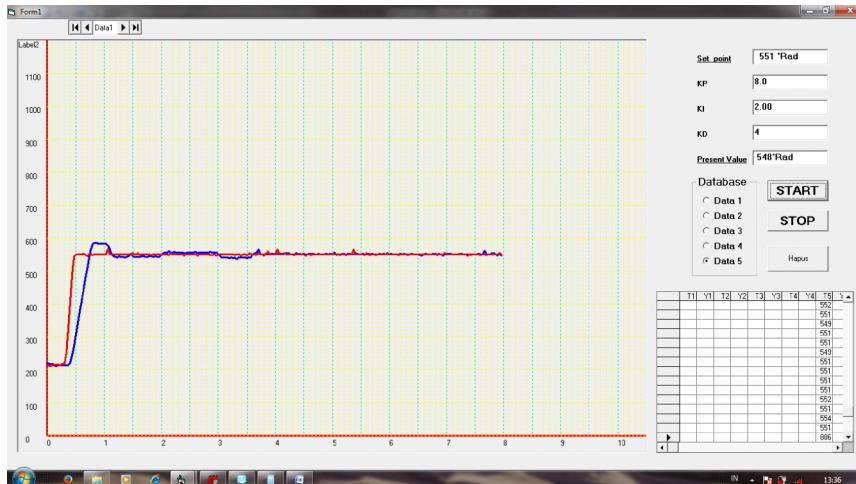
Gambar 16. Kurva pengendalian posisi motor servo dengan parameter $K_p = 10$ $K_i = 0.5$ $K_d = 4$



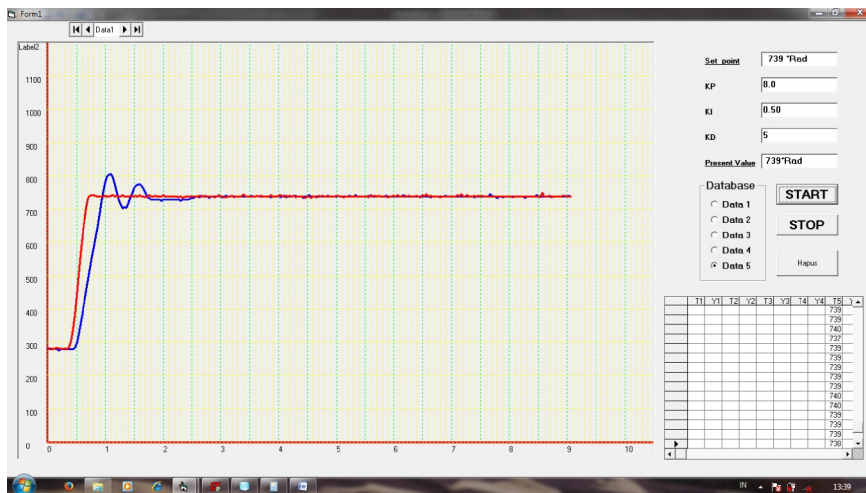
Gambar 17. Kurva pengendalian posisi motor servo dengan parameter $K_p = 15$ $K_i = 0.5$ $K_d = 4$



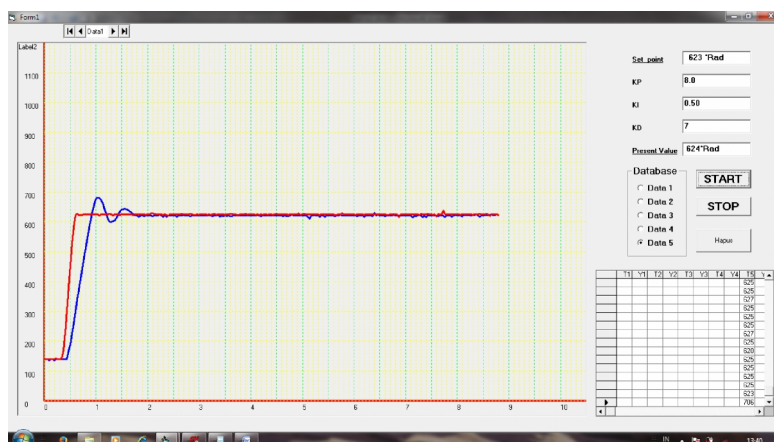
Gambar 18. Kurva pengendalian posisi motor servo dengan parameter $K_p = 8$ $K_i = 0.30$ $K_d = 4$



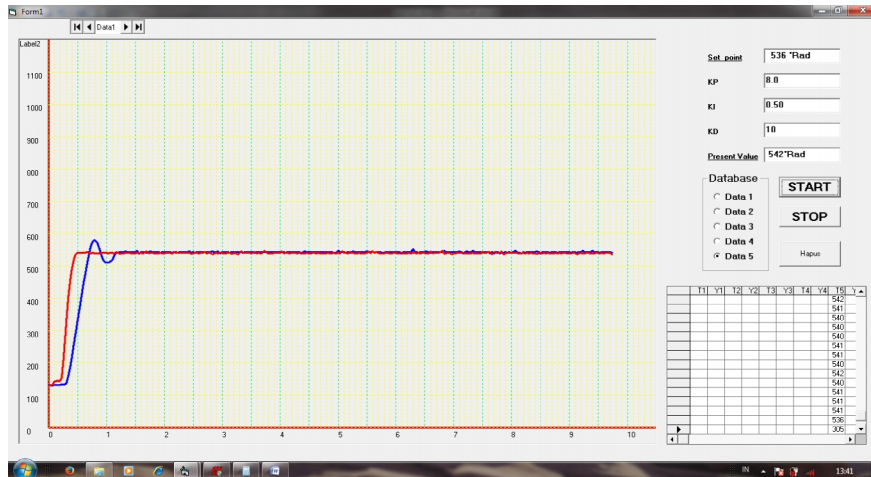
Gambar 19. Kurva pengendalian posisi motor servo dengan parameter $K_p = 8$ $K_i = 2.00$ $K_d = 4$



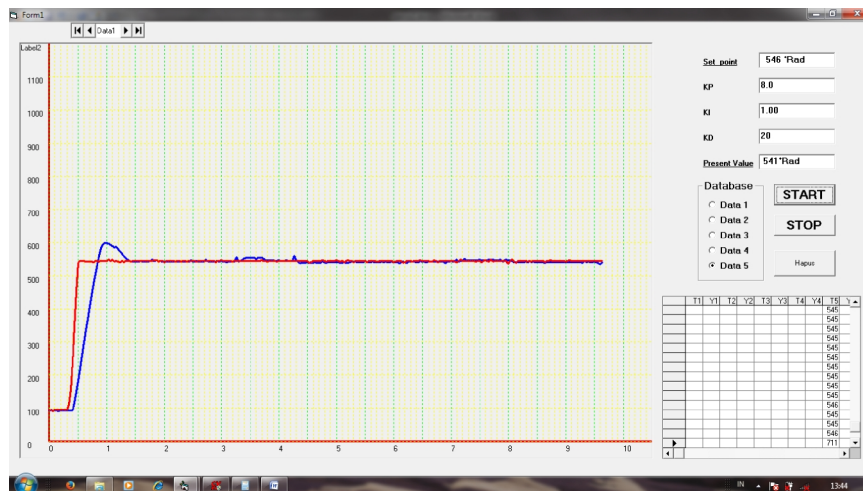
Gambar 20. Kurva pengendalian posisi motor servo dengan parameter $K_p = 8$ $K_i = 0.50$ $K_d = 5$



Gambar 21. Kurva pengendalian posisi motor servo dengan parameter $K_p = 8$ $K_i = 0.50$ $K_d = 7$



Gambar 22. Kurva pengendalian posisi motor servo dengan parameter $K_p = 8$ $K_i = 0.50$ $K_d = 10$



Gambar 23. Kurva pengendalian posisi motor servo dengan parameter $K_p = 8$ $K_i = 1.00$ $K_d = 20$

4. KESIMPULAN

Pada penelitian dan pengujian ini masih merupakan prototype untuk merealisasikan konsep pengendalian dengan kendali PID. Dari hasil penelitian ini telah membuktikan konsep kendali PID untuk diterapkan pada kendali motor servo untuk mencapai posisi yang diinginkan dengan kecepatan respon dengan mengubah nilai parameter K_p , K_i dan K_d . Dari hasil perancangan dan uji coba kendali motor servo posisi dengan kendali mikrokontroler dapat disimpulkan sebagai berikut :

1. Kecepatan mencapai keadaan stabil (Steady State) tergantung pada nilai parameter K_p , K_i , dan K_d .
2. Penentuan kombinasi K_p , K_i dan K_d akan menghasilkan kecepatan mencapai nilai steady state dengan simpangan nilai simpangan terkecil.
3. Dari hasil pengujian nilai optimum kurva pengendalian posisi motor servo pada parameter $K_p = 7$ $K_i = 0.5$ $K_d = 4$
4. Penelitian ini masih berupa pembuktian konsep kendali PID yang direalisasikan pada alat yang akan digunakan sebagai peraga pembelajaran pada mata kuliah kendali. Selanjutnya untuk penggunaan konsep kendali PID pada penerapan pada fungsi yang lebih spesifik fungsi aplikasi lanjut akan dilakukan pada penelitian selanjutnya.

REFERENSI

- Atmel. (2003). *8-bit Microcontroller with 32K Bytes In-System Programmable Flash*, <http://www.atmel.com/product/AVR/> .
- Agus,B. (2008), *C dan AVR: Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATmega8535*. Yogyakarta. Graha Ilmu.
- Gopal, M. (2003). *Control Systems*. USA. McGraw-Hill.
- Setiawan, I. (2008). *Kendali PID untuk Proses Industri*. Jakarta. PT. Elex Media Komputindo.