

ROBOT LENGAN 3 DOF DENGAN INPUT SINYAL EMG

Andreas Widya Saputra, Paulus Susetyo W, ST, Ir. Rika Rokhana, M.T, Ir. Kemalasari, M.T
Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya
Kampus PENS-ITS Sukolilo, Surabaya

Abstrak – Otot adalah bagian tubuh manusia yang berfungsi dalam sistem gerak. *Electromyography* berfungsi untuk mendeteksi adanya potensial listrik yang dihasilkan pada saat otot kontraksi dan relaksasi, sinyal tersebut digunakan untuk menggerakkan lengan robot buatan, dimana nantinya robot lengan ini bias digunakan dalam membantu manusia untuk mencapai tempat-tempat yang berbahaya dan kuran bersahabat dengan manusia.

I. PENDAHULUAN

1.1 Latar Belakang

Robotika merupakan ilmu yang sedang berkembang di dunia, penggunaannya tidak terbatas hanya pada dunia industri saja, tapi juga sudah merambah di sekolah-sekolah dan perguruan tinggi di Indonesia. Kemajuannya yang pesat dan sisi menguntungkan, akurasi dan kecepatan serta menghibur dari sisi robotika membuat ketertarikan banyak pihak, salah satunya adalah pihak medis, dimana banyak sekarang ditemui alat-alat robotika ditemui dalam dunia kedokteran, misalnya alat operasi menggunakan robot, dan sebagainya.

Pada proyek akhir ini menyangkut juga sisi medis yaitu sinyal EMG (*electromyogram*) / sinyal otot yang diambil pada tangan kemudian dilakukan pengekstraksian sehingga didapat suatu bentuk sinyal yang lebih mudah diolah, untuk kemudian sinyal tersebut akan digunakan untuk menggerakkan robot lengan 3 DOF yang berbentuk menyerupai tangan, dimana diharapkan ini dapat membantu untuk mengembangkan kemajuan IPTEK dan juga mungkin membantu dalam dunia industri dalam mengganti tangan manusia untuk menjangkau tempat-tempat yang tidak bisa/membahayakan manusia.

1.2 Tujuan

Tujuan utama dari proyek akhir ini adalah merencanakan dan merealisasikan sebuah lengan robot 3 DOF yang digerakkan oleh sinyal EMG. Mengacu pada tujuan utama pada proyek akhir ini maka terdapat beberapa tujuan khusus antara lain :

1. Merancang dan membuat sistem komunikasi untuk proses pengiriman data dalam sistem komunikasi antar PC dan mikrokontroler yang digunakan.
2. Merancang dan membuat lengan robot bergerak sesuai dengan arah yang sesuai dengan input sinyal EMG yang masuk.
3. Motor berputar konstan di semua posisi.

1.3 Rumusan Masalah

Adapun permasalahan yang akan dibahas adalah sebagai berikut :

1. Bagaimana membuat robot lengan agar posisi dan arahnya sesuai dengan input sinyal EMG yang dimasukkan.
2. Bagaimana membuat kontroler dapat berkomunikasi dengan PC sebagai sumber data.
3. Bagaimana membuat motor dc bergerak dengan steady state yang cepat dan membuat motor berputar dalam kecepatan konstan di segala posisi.

1.4 Batasan Masalah

Adapun batasan-batasan masalah yang dibuat agar dalam pengerjaan proyek akhir ini dapat berjalan dengan baik adalah sebagai berikut :

1. Gerak robot terbatas sesuai dengan gerakan lengan tangan manusia yang ditentukan, yaitu hanya berupa gerakan *fleksi* dan *ekstensi*, *abduksi* dan *adduksi*, *elevasi* dan *depresi*.
2. Sendi bergerak satu per satu (tidak bersamaan)

1.5 Metodologi

Metodologi dalam pembuatan proyek akhir ini meliputi :

1.5.1 Pemahaman Materi

Pada tahap ini dilakukan upaya memahami materi dari beberapa literatur yang digunakan baik berupa buku, website atau jurnal ilmiah tentang PID, komunikasi serial, bahasa pemrograman C++ (codevision AVR) dan lain-lain yang dapat membantu penyelesaian proyek akhir ini.

1.5.2 Perancangan mekanik

Pada tahap ini yang dilakukan adalah membuat perancangan mekanik robot. Serta diupayakan mempunyai ukuran yang sama dengan lengan aslinya.

1.5.3 Pembuatan Program

Pada tahap ini yang dilakukan adalah membuat program yang memanfaatkan PID untuk menghasilkan satu keputusan aksi kontrol motor dc sebagai aktuator berdasarkan keputusan aksi kontrol motor dc sebagai aktuator berdasarkan input data dari PC yang merupakan ekstraksi dari sinyal otot / EMG, serta berusaha membuat kecepatannya konstan.

1.5.4 Pengujian Sistem

Pada tahap ini dilakukan pengujian pada motor dc dan sensor kecepatan. Setelah itu dilakukan pengujian program dengan memasukkan data input. Program PID untuk mempertahankan kecepatan motor secara konstan di berbagai posisi.

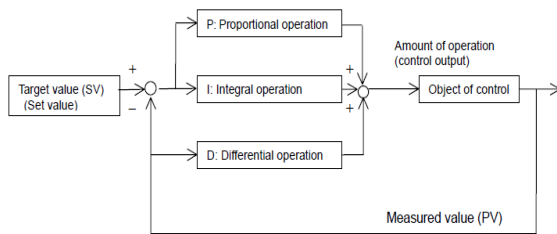
1.5.5 Analisa Hasil Program

Pada tahap terakhir ini dilakukan analisa terhadap hasil yang ada yang meliputi analisa program dan hasil respon dari PID yang telah dibuat dibandingkan dengan gerak tangan yang sebenarnya.

II. TINJAUAN PUSTAKA

2.1 Berikut adalah teori-teori penunjang yang digunakan untuk menyelesaikan tugas akhir ini antara lain:

2.1 PID



Gambar 2.1. Konsep dasar logika fuzzy

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran system terhadap masukan tertentu sebagaimana yang diinginkan.

1. Kontrol Proporsional

Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) * e$ maka $u = K_p * e$ dengan K_p adalah Konstanta Proporsional. K_p berlaku sebagai Gain (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya rise time dan settling time.

2. Kontrol Integratif

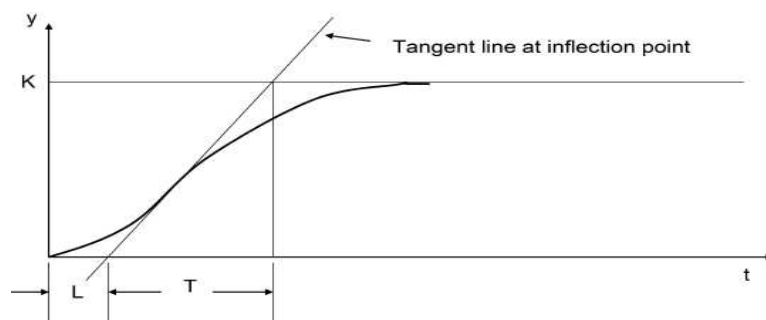
Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = [\int e(t) dt] K_i$ dengan K_i adalah konstanta Integral, dan dari persamaan diatas, $G(s)$ dapat dinyatakan sebagai $u = K_i \cdot [\Delta e / \Delta t]$ Jika $e(T)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika $e(T)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde sistem

3. Kontrol Derivatif

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s) = s \cdot K_d$ Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini dalam konteks "kecepatan" atau rate dari error. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri

Tunning PID dengan metode Ziegler-nichols

Metode 1 dari Ziegler-Nichols. Kurva berbentuk s hasil respon motor dapat dicirikan oleh dua konstanta, waktu tunda L dan waktu konstan T, waktu yang ditentukan dengan menggambar garis singgung pada titik belok dari kurva dan menemukan persimpangan dari garis singgung dengan sumbu waktu dan garis tingkat daerah stabil



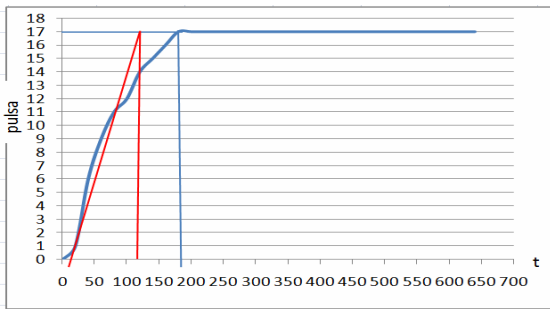
Gambar 2.2. Kurva reaksi

Untuk rule perhitungan konstanta K_p , K_i dan K_d dengan metode Ziegler–Nichols adalah sebagai berikut:

Tabel 2.1. Tabel rule Ziegler-Nichols tuning

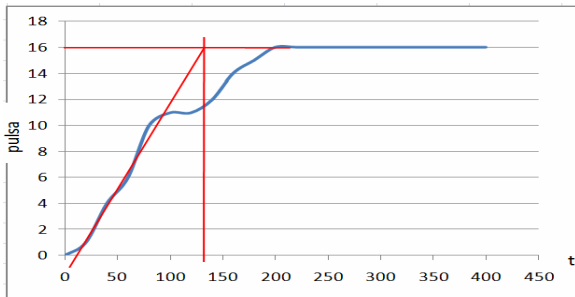
Controller	K_P	K_I	K_D
P	$\frac{T}{L}$	0	0
PI	$0.9 \frac{T}{L}$	$0.27 \frac{T}{L^2}$	0
PID	$1.2 \frac{T}{L}$	$0.6 \frac{T}{L^2}$	$0.6T$

Dengan mengetahui kurva s dari karakteristik motor kita dapat menentukan konstanta K_p , K_i dan K_d dengan rule-rule di atas setelah kita mendapati nilai L dan T dari kurva motor



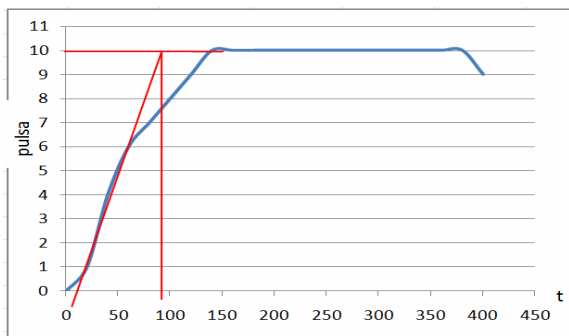
Gambar 2.3 kurva karakteristik motor 1

$K_p = 8.55086$ $K_i = 305.388$ dan $K_d = 0.05986$



Gambar 2.4 kurva karakteristik motor 1

$K_p = 9.63429$, $K_i = 344.082$ dan $K_d = 0.06744$.

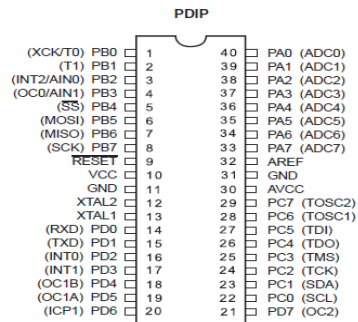


Gambar 2.5 kurva karakteristik motor 1

$K_p = 6.96738$, $K_i = 267.976$ dan $K_d = 0.04592$.

2.2 MIKROKONTROLLER ATMEGA32

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial USART, programmable Watchdog Timer, dan mode power saving. Mempunyai ADC dan PWM internal. AVR juga mempunyai In-System Programmable Flash on-chip yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. ATmega32 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan.



Gambar 2.6 kurva karakteristik motor 1

Pada mikrokontroler AVR Atmega 32 dapat diberikan sinyal pulsa hingga 16 MHz sehingga dapat mengeksekusi program dengan cepat sehingga didapatkan sistem yang *real time*. Selain itu, pada mikrokontroler ini juga terdapat *Flash Programming* sebesar 2 Kbyte.

Timer

Timer/Counter adalah salah satu peripheral yang ada pada mikrokontroler. PWM (*Pulse Width Modulation*) adalah salah satu mode dari beberapa metode konversi digital ke analog. PWM diatur dengan *duty cycle* gelombang kotak dari keluaran mikrokontroler berupa tegangan DC.

Pada mikrokontroler ATMEGA 32, Timer pada mode PWM pada Timer 1 penghitungan dimulai dari bawah yaitu 0 hingga ke nilai tertinggi kemudian kembali ke 0 lagi. Nilai teratas diatur oleh resolusi yang terdapat pilihan 8-bit, 9-bit dan 10-bit. Resolusi dari WGM (*Waveform Generator Mode*).

Register timer pada mikrokontroler

Bit - 7	Bit - 6	Bit - 5	Bit - 4	Bit - 3	Bit - 2	Bit - 1	Bit - 0
COM 1A1	COM 1A0	COM 1B1	COM 1B0	FO C1A	FO C1B	WG M11	WG M10

Gambar 2.7 Register timer pada mikrokontroler

Tabel 2.1 Mode pelihan Gelombang PWM

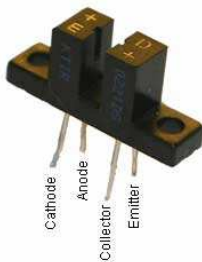
WGM 11	WGM 10		
0	0	PWM Disabled	
0	1	8 – bit	255 (0xff)
1	0	9 – bit	511 (0x1ff)
1	1	10 – bit	1023 (0x3ff)

USART

Pada mikrokontroler ATMEGA 32 menggunakan serial komunikasi USART. Sehingga Data Rx berfungsi sebagai pengirim dan penerima data sedangkan Tx sebagai output clock. Output dari serial pada mikrokontroler ATMEGA 32 berupa tegangan TTL 0 – 5 Volt.

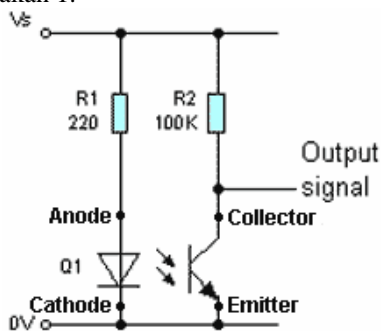
2.3 OPTOCOUPLER-U

Optocoupler ini terdiri dari infrared LED dan phototransistor yang dibungkus menjadi satu wadah, seperti gambar 3, phototransistor diatur sedemikian mungkin sehingga dapat mendeteksi infrared dari LED



Gambar 2.8 OPTOCOUPLER-U

Lalu untuk cara kerjanya sendiri adalah bila ada sebuah benda diletakkan di antara LED dan phototransistor sehingga cahaya infrared tidak tertangkap oleh phototransistor maka output akan 0 sedang bila tidak ada benda/penghalang yang menghalangi infrared agar tertangkap oleh phototransistor maka output akan 1.



Gambar 2.9 Gambar rangkaian Optocoupler-u

2.4 PWM

Dalam kasus pengendalian kecepatan putar motor, tegangan terminal motor (V) merupakan variabel yang dapat diatur untuk menghasilkan putaran yang diinginkan. Dalam penelitian ini pengaturan besarnya dilakukan dengan proses Chopper (pengaturan PWM). Besarnya tegangan output dinyatakan sebagai :

$$V_{out} = \frac{T_{on}}{T_{off} + T_{on}} \times V_{in}$$

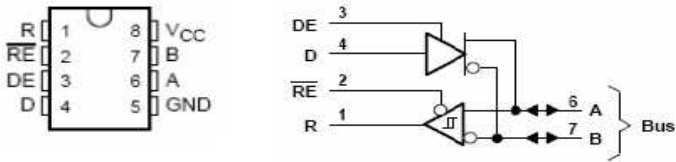
2.5 Komunikasi data serial

Komunikasi data serial sangatlah berbeda dengan format pemindahan data paralel. Disini, pengiriman bit-bit tidak dilakukan sekaligus melalui saluran paralel, tetapi setiap bit dikirimkan satu persatu melalui saluran tunggal. Dalam pengiriman data secara serial harus ada sinkronisasi atau penyesuaian antara pengirim dan penerima agar data yang dikirimkan dapat diterima dengan tepat dan benar oleh penerima. Dalam komunikasi secara serial terdapat tiga macam mode transmisi serial dalam mentransmisikan bit-bit data, yaitu : *synchronous*, *asynchronous* dan *isochronous* saja yang digunakan. Transmisi serial mode *asynchronous* digunakan bila pengiriman data dilakukan satu karakter tiap pengiriman. Antara satu karakter dengan yang lainnya tidak ada waktu yang tetap. Karakter dapat dikirimkan sekaligus ataupun beberapa karakter kemudian berhenti untuk waktu yang tidak tertentu, kemudian dikirimkan sisanya. Dengan demikian bit-bit data ini dikirimkan dengan periode yang acak sehingga pada sisi penerima data akan diterima kapan saja. Adapun sinkronisasi yang terjadi pada transmisi serial *asynchronous* adalah dengan memberikan bit-bit pertanda awal dari data dan penanda akhir dari data pada sisi pengirim maupun dari sisi penerima.

2.6 RS 485

Dengan menggunakan RS 485 komunikasi data dapat dilakukan pada jarak yang cukup jauh yaitu 1,2 Km. Selain dapat digunakan untuk jarak yang jauh teknik ini juga dapat digunakan untuk menghubungkan 32 unit beban sekaligus hanya dengan menggunakan dua buah kabel saja tanpa memerlukan referensi *ground* yang sama antara unit yang satu dengan unit lainnya.

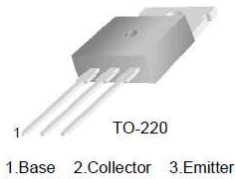
MAX 485 adalah IC yang menjadi komponen utama Modul SR-485 yang didisain untuk komunikasi data secara bidirectional atau *multipoint* dengan Standard ANSI EIA/TIA-422-B dan ITU V11. Data yang ditransmisikan oleh IC ini dikirim dalam bentuk perbedaan tegangan yang ada pada kaki A dan B dari MAX485



Gambar 2.10 Konfigurasi pin max 485

MAX485 berfungsi sebagai pengirim data atau penerima data tergantung dari kondisi kaki-kaki kontrolnya yaitu DE dan RE. Apabila kaki DE berlogika 0 dan RE berlogika 0, maka MAX485 berfungsi sebagai penerima data sedangkan bila kaki DE berlogika 1 dan RE berlogika 1 maka MAX485 berfungsi sebagai pengirim.

2.7 TIP 122 dan TIP 127

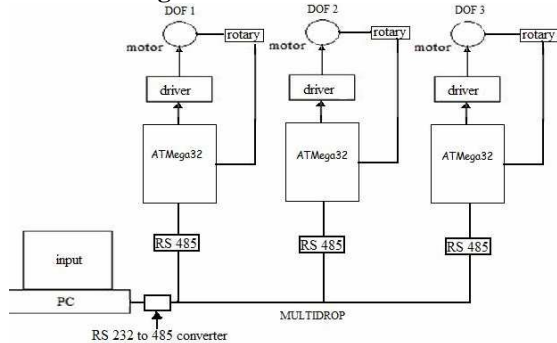


Gambar 2.11 bentuk fisik transistor TIP

TIP 122 merupakan transistor NPN sedang TIP 127 adalah transistor PNP keduanya digunakan untuk menghasilkan suatu rangkaian h-bridge sebagai driver motor. Pemilihan jenis ini dikarenakan karena arus I_c yang mampu sampai dengan 5A (dari datasheet) sehingga untuk kebutuhan tugas akhir ini dengan beban motor 1-2A dirasa sangat cukup.

III. PERANCANGAN DAN PEMBUATAN ALAT

3.1 Perancangan Sistem



Gambar 3.1. Blok Diagram Sistem

3.2 Rangkaian Instrumentasi

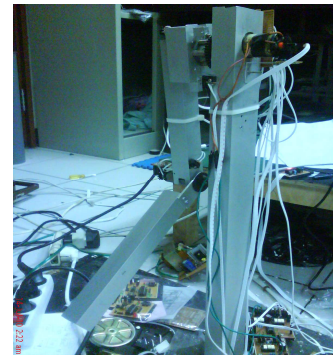
Perancangan sistem dari proyek akhir yang berjudul "ROBOT LENGAN 3 DOF DENGAN INPUT SINYAL EMG" ini mempunyai konfigurasi blok diagram sistem seperti gambar 3.1 dengan alur kerja plant / sistem sebagai berikut :

1. PC akan mengirim paket data yang berisi kondisi motor1, motor2 dan motor 3 ke mikrokontroler.
2. Mikrokontroler Atmega 32 akan menerima data tersebut dan mencocokkan protocolnya apakah data tersebut untuk mikrokontroler tersebut.
3. Motor akan berputar atau berhenti sesuai yang diperintahkan mikrokontroler.
4. Dengan berputarnya motor maka rotary encoder yang terpasang pada as motor akan berputar juga, untuk kemudian dengan optocoupler (sensor kecepatan) didapati kecepatan motor tersebut.
5. Dengan data kecepatan tersebut maka kontrol PID akan berusaha mengkoreksi error dan mencoba untuk mencapai kecepatan yang diinginkan (referensi).

Dari sistem tersebut diketahui bahwa sistem terintegrasi antar mekanik, aktuator, sensor dan controller. Sehingga setiap adanya hambatan atau kesalahan pada salah satu bagian tersebut membuat sistem tidak dapat berjalan dengan semestinya.

3.2 PERANCANGAN MEKANIK

Perancangan mekanik pada sistem ini, didasarkan pada ukuran lengan manusia yang sesungguhnya. Keseluruhan mekanik robot menggunakan bahan aluminium, karena bahan-bahan tersebut cocok untuk digunakan dalam mendesain robot, karena ringan tapi kuat dan mudah dibentuk. Untuk meminimalisir bobot robot, PCB hardware akan diletakkan terpisah dari lengan robot.



Gambar 3.1. bentuk mekanik

3.3 PERANCANGAN HARDWARE

Hardware pada sistem ini menggunakan beberapa perangkat atau komponen yaitu :

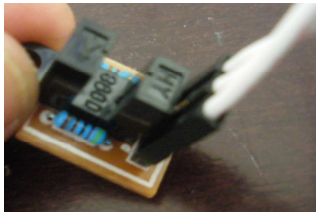
3.3.1 Minimum System

Pada sistem pengontrol utama ini menggunakan Atmega 32 untuk melakukan komunikasi dengan PC, parsing data maupun untuk melakukan menjalankan piranti-piranti lain serta membaca sensor. Atmega 32 dipilih sebagai controller tiap dof karena dirasa cukup, karena memiliki 2

buah timer (dalam hal ini dipakai sebagai counter dan PWM), serta beberapa pin (dalam tugas ini dibutuhkan hanya 4 buah pin, yaitu: enable motor, direction motor, rem, dan limit switch). Data dari PC akan diolah apakah data yang dikirim untuk minimum sistem tersebut atau untuk minimum sistem yang lain dengan menggunakan protokol.

3.3.2 Sensor Kecepatan

Dari rotary encoder yang terdapat pada mouse bola ditambah dengan optocoupler-u maka akan didapatkan sensor kecepatan.



Gambar 3.2. sensor kecepatan

3.4 PERANCANGAN SOFTWARE

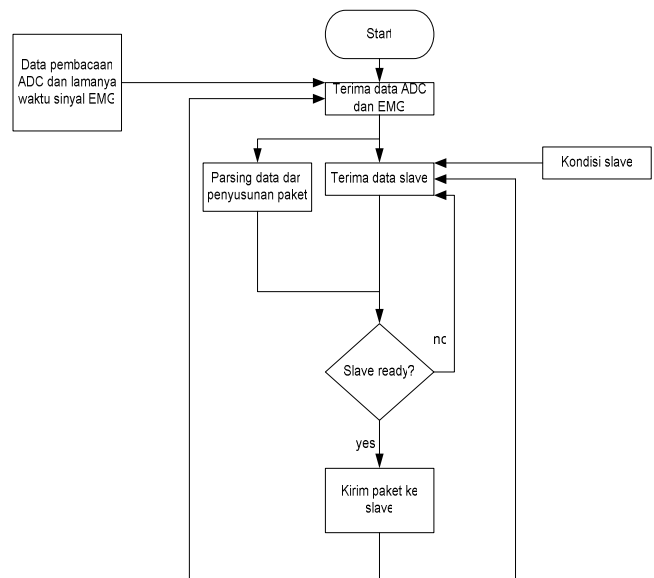
Perancangan software dari sistem ini dibagi menjadi beberapa bagian utama yaitu software komunikasi, software sistem PID, software pembacaan kecepatan dan software PWM. Berikut ini adalah program-program yang terdapat dalam sistem

3.4.1 Perancangan Software Komunikasi

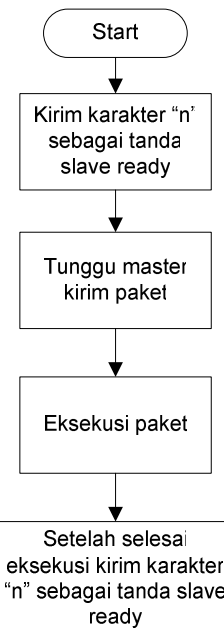
Dalam membuat software komunikasi ini menggunakan USART yang terdapat pada Atmega 32 yang akan berkomunikasi dengan PC.

Pada sub program tersebut, terdapat pengiriman data untuk menentukan motor mana yang akan berputar atau berhenti dari PC ke mikrokontroler.

Setelah didapatkan hasil parsing data maka akan dicek mana motor yang harus berputar lalu kemudian memerintahkan / memberi logika yang sesuai ke driver motor tersebut.



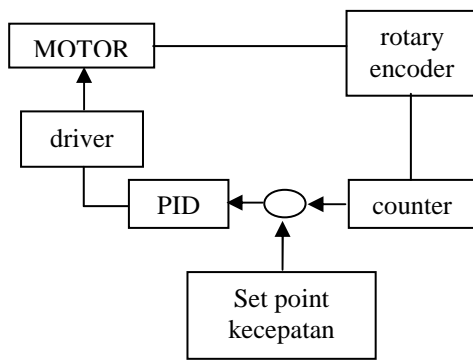
Gambar 3.2. Algoritma master



Gambar 3.4 Algoritma Slave

3.4.2 Perancangan Kontrol PID

Kontrol PID yang dibuat adalah dalam bentuk software sehingga dapat memperringkas hardware serta mempermudah dalam menentukan koefisien PID tersebut. Kontrol PID dalam sistem ini digambarkan sebagai berikut



Gambar 3.3 Blok diagram Kontrol PID

Dari blok diagram diketahui bahwa kontrol PID didapatkan dari umpan balik langsung dari sensor kecepatan yang terpasang pada as motor dc kemudian akan dihitung besarnya kecepatan untuk kemudian dikoreksi errornya dan berusaha mencapai kecepatan yang dikehendaki (referensi).

Untuk program kontrol PID sendiri juga menggunakan error sebelum-sebelumnya untuk menentukan nilai PID berikutnya yang menjadi dasar koordinat obyek berikutnya. Berikut ini adalah flowchart dan program kontrol PID yang terdapat dalam sistem ini

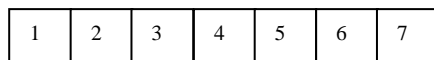
3.4.3 Perancangan software pembacaan kecepatan

Untuk pembacaan kecepatan dilakukan dengan memasukkan data tersebut kedalam salah satu timer pada Atmega 32, untuk kemudian jumlah pulsa tersebut dihitung dan dikomparasi dengan jumlah pulsa referensi.

3.4.4 Perancangan PWM

Untuk PWM bisa diambilkan dari pin timer juga, dimana nanti akan dipilih mode PWM sehingga dihasilkan PWM yang dikehendaki.

Untuk *slave* cara kerjanya adalah pertama kali *slave* mengirimkan karakter "n" sebagai tanda *slave* ready, untuk kemudian menunggu visual basic mengirimkan paket datanya. Setelah eksekusi maka *slave* akan mengirimkan lagi karakter "n" sebagai tanda siap untuk menerima data lagi. Paket datanya adalah sebagai berikut:



Gambar 3.15 Paket data

1 paket berisi 7 karakter, dimana perincan untuk setiap karakter adalah sebagai berikut:

- Karakter 1: Header
 " #" untuk *slave* 1, "@" untuk *slave* 2,
 "!" untuk *slave* 3.

Karakter 2: Arah lengan
 "1" untuk naik, "2" untuk turun dan "0" untuk stop.

Karakter 3,4,5,6: lamanya lengan bergerak.
 (misal : "2000" berarti 2000ms)

Karakter 7: Tail
 "x" untuk *slave* 1, "y" untuk *slave* 2, "z" untuk *slave* 3.

Jadi dari blok diagram diatas, saat dimulai maka minimum sistem akan membaca data yang masuk (data dari PC) untuk kemudian akan dilihat apa motor ada yang jalan atau tidak, jika tidak ada maka akan discan lagi sampai ditemui ada perintah untuk motor run, setelah didapati ada perintah motor run maka kemudian akan discan kecepatan motor tersebut, untuk kemudian data kecepatan akan masuk kembali ke atmega 32 untuk kemudian diolah dengan PID dimana bila didapati kecepatan tidak sesuai maka PID akan berusaha untuk mencapai nilai referensi yang diberikan.

Konfigurasi PIN yang digunakan:

- PIN PA.5,PA.6 —digunakan untuk pengatur arah dan rem motor
- PIN PB.0 —digunakan untuk counter kecepatan
- PIN PA.7 —digunakan untuk mengatur PWM
- PIN PD.4,PD.5 —digunakan untuk limit switch

IV. PENGUJIAN DAN ANALISA

Pada bab ini akan dilakukan pengujian dan analisa terhadap kinerja mekanik dan kendali PID yang telah dirancangkan dibuat pada bab sebelumnya. Untuk tahapan-tahapan pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian mekanik lengan robot
 - Pengujian motor
 - Pengujian rotary encoder sebagai sensor kecepatan
2. Pengujian respon motor dengan PID.
3. Pengujian komunikasi multidrop.

4.1 PENGUJIAN MEKANIK ROBOT LENGAN

Pada tahap ini akan dilakukan pengujian terhadap kinerja tiap-tiap motor dc yang digunakan sebagai penggerak lengan robot baik dengan kondisi tanpa atau dengan efek pembebanan.

Dilakukan pengujian terhadap kinerja motor untuk bisa mengetahui besar torsi dan tegangan yang dibutuhkan untuk menggerakkan lengan robot sehingga dapat mencapai kecepatan yang sama di tiap posisi.

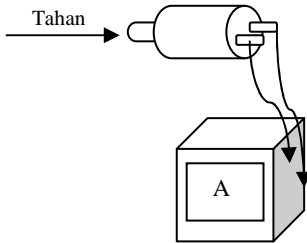
4.1.1 Pengujian motor tanpa dan dengan efek pembebanan

1) Tujuan

- Untuk mengetahui besar arus serta tegangan untuk bisa menggerakkan lengan tanpa dan dengan beban.
- Untuk dapat mengetahui besar torsi serta gaya yang bekerja pada lengan robot tersebut.

2) Parameter yang digunakan

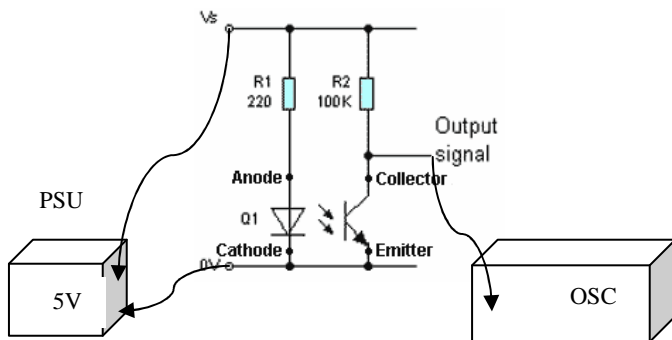
- Tegangan Input = $\pm 9V$
 - Arus input = $\pm 1-2A$
- 3) Setting pengujian
- Untuk mengetahui besar tegangan dan arus yang dibutuhkan motor untuk menggerakkan lengan dengan kondisi ditahan.
 - Konfigurasi sistem diuji tampak pada gambar 4.1



Gambar 4.1 pengujian tegangan dan arus kerja motor

4.1.2 Pengujian pembacaan rotary encoder sebagai sensor kecepatan

- 1) Tujuan
- Tujuan dari pengujian ini adalah untuk mendapat suatu nilai untuk digunakan sebagai nilai referensi kontrol PID sebagai pembanding untuk mendapatkan nilai error dan delta error yang masuk pada rumus PID.
- 2) Setting Pengujian
- Untuk pengujian ini rangkaian optocoupler yang terdiri dari 3 pin keluaran yaitu data, + dan -, dimana pin + dan - dihubungkan ke PSU dan diberi tegangan 5V, kemudian pin data masuk pada oscilloscope.



Gambar 4.2 posisi pengujian sensor kecepatan

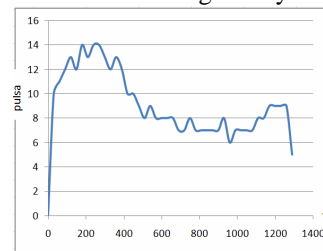
- 3) Pengujian yang dilakukan
- Pasang optocoupler pada as motor yang telah dipasang rotary encoder.
 - Beri tegangan 5V pada pin keluaran + dan - sensor
 - Amati hasilnya pada oscilloscope



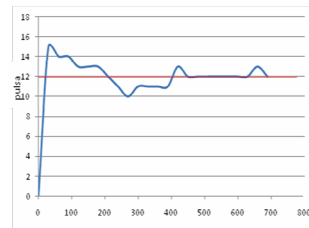
Gambar 4.3 hasil pengujian sensor rotary encoder

4.2 Pengujian Sistem kendali PID

- 1) Tujuan
- Tujuan dari pengujian sistem kendali ini adalah mendapatkan setting point kecepatan awal motor dengan berusaha mengurangi / mempercepat waktu steady state pada tiap motor.
- 2) Parameter yang digunakan
- Untuk melihat pengaruh yang diberikan oleh kontroler terhadap plant maka dapat dilihat pada perubahan respon motor yang kemudian dibandingkan dengan respon tanpa kontroler (open loop). Pengujian respon motor yang dipergunakan dalam tugas akhir ini menggunakan data dari sensor yang kemudian masuk dalam timer untuk di hitung di mikrokontroler untuk kemudian hasilnya dikirim ke laptop melalui komunikasi serial, kemudian nilai-nilai tersebut dimasukkan dalam excel untuk kemudian dibuat grafiknya

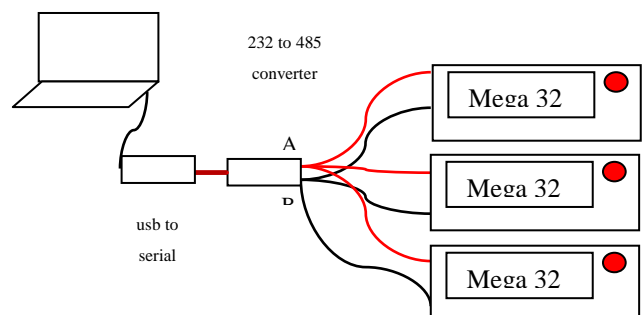


Setelah PID:



4.3 Pengujian komunikasi multidrop.

- 1) Tujuan
- Tujuan dari pengujian ini adalah untuk memastikan bahwa baik minimum sistem maupun 232 to 485 converter dan protokol komunikasi berjalan lancar.
- 2) Setting pengujian
- Pasang output pin PORTA.7 ke port yang tersambung pada led yang terdapat pada minimum sistem, selanjutnya download program.
- 3) Pengujian yang dilakukan



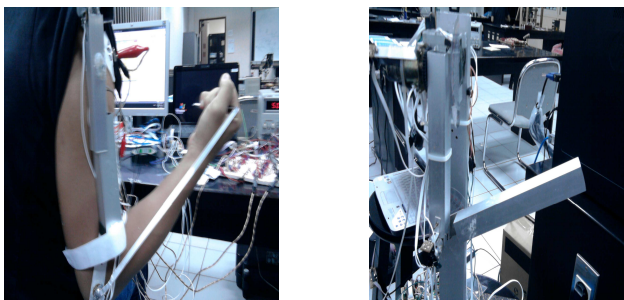
Gambar 4.17 Pengujian komunikasi

Pada visual basic akan mengirim paket data, dimana ada 5 bit karakter setiap pengiriman yang terdiri dari protokol+time nyala led+ tail. Minsis 1 kita beri header "# dan tail "x", minsis 2 kita beri header "@" dan tail "y", dan minsis 3 kita beri header "!" dan tail "z". Pada visual basic dikirim data berupa "#2000x" dan yang terjadi adalah minsis 1 lednya menyala selama kurang lebih 2 detik (pada chip mega32

4.4 Pengujian integrasi

Lengan 1

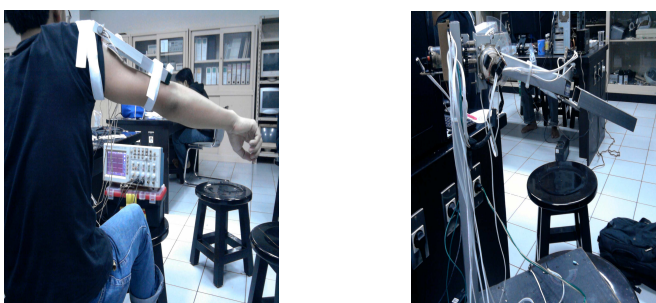
Time	Robot (°)	Manusia (°)	Error %
756	88	90	2,2
840	90	85	5,8
1401	118	115	2,6



Gambar 4.30 Pengujian posisi lengan 1 manusia dan robot

Lengan 2

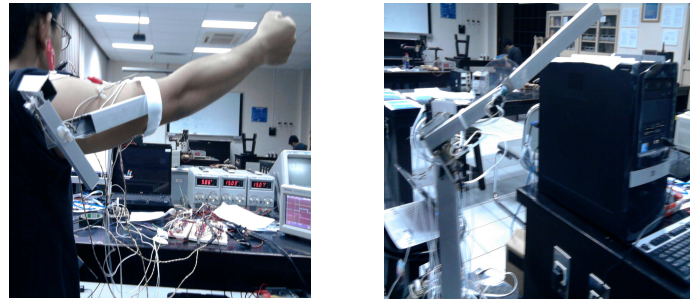
Time	Robot (°)	Manusia (°)	Error %
1401	75	80	6.25
1207	72	75	4
1360	85	83	2.4



Gambar 4.31 Pengujian posisi lengan 2 manusia dan robot

Lengan 3

Time	Robot (°)	Manusia (°)	Error %
1685	98	95	3,1
1992	122	120	1,6
1933	115	110	4,5



Gambar 4.32 Pengujian posisi lengan 3 manusia dan robot

BAB V PENUTUP

5.1 Kesimpulan

Dari proyek akhir ini dapat diambil kesimpulan bahwa:

1. Kecepatan motor yang diatur konstan belum bisa menyamai posisi lengan sebenarnya, dikarenakan lengan dapat mencapai suatu sudut yang sama dengan waktu yang tercatat pada visual basic berbeda-beda, sedangkan robot dapat mencapai suatu sudut dengan waktu yang sama yang sudah dihitung sebelumnya. Error maksimal pada pengujian lengan pertama untuk integrasi adalah 2.6 %, untuk lengan 2 sebesar 6.25 % dan lengan 3 adalah 4.5%.
2. Motor bekerja pada tegangan 9v, dan dengan pemasangan lengan untuk mendapatkan kecepatan yang hampir sama (pengamatan secara visual) dengan kecepatan tangan bergerak maka tegangan yang dibutuhkan adalah 18v, dengan *set point/10ms* untuk gerak lengan naik motor 1 adalah 6 dan untuk turun adalah 7, untuk motor 2 *set point/10ms* untuk gerak lengan naik adalah 15 dan turun dengan *set point/10ms* adalah 15 dan untuk turun adalah 12.
3. Dari pengujian kendali PID didapati bahwa PID dapat memperkecil *rise time* saat lengan terpasang, untuk motor 1 *rise time* tanpa kontrol adalah 16 ms, dengan PID maka *rise time* menjadi 10 ms. Untuk motor 2 *rise time* tanpa kontrol adalah 30 ms, dengan PID maka *rise time* menjadi 10 ms. Dan motor 3 *rise time* tanpa kontrol adalah 15 ms, dengan PID maka *rise time* menjadi 10 ms.
4. Komunikasi secara *multidrop* dengan menggunakan max485 bisa berjalan dengan baik sesuai program dan protocol yang diberikan.

5.2 Saran

Untuk pengembangan tahap-tahap berikutnya dapat menambah untuk pengolahan sinyal EMG dengan menampilkan powernya, sehingga mungkin bisa menjadi acuan sebagai nilai *set point variable* (dapat berubah-ubah).

DAFTAR PUSTAKA

- [1] ca.digikey.com/1/3/indexb1038.html
- [2] en.wikipedia.org/wiki/PID_controller
- [3] <http://www.alldatasheet.com/datasheet-pdf/pdf/239668/TAITRON/TIP122.html>
- [4] <http://www.alldatasheet.com/datasheet-pdf/pdf/239668/TAITRON/TIP127.html>
- [5] Katsuhiko Ogata, *Modern Control Engineering*, Prentice hall, 1997.
- [6] oc.its.ac.id/ambilfile.php?idp=309
- [7] Wolfgang Altmann, Dave Macdonald, *Practical Process control for Engineering and Technicians*, IDC Technologies, 2005.
- [8] www.airborn.com.au/layout/232to485.html
- [9] www.atmel.com/dyn/resources/prod_documents/2503s.pdf
- [10] www.futurlec.com/Maxim/MAX485.shtml