

PENGENDALI POSISI BERBASIS PC

Laporan tugas akhir ini diajukan sebagai salah satu syarat memperoleh gelar Ahli
Madya (A.Md)

OLEH

HARBHANU PRIMA SALOKHA

2304220648



PROGRAM STUDI DIPLOMA 3 INSTRUMENTASI INDUSTRI DAN ELEKTRONIKA

DEPARTEMEN FISIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS INDONESIA

DEPOK

2008

LEMBAR PENGESAHAN

Nama : Harbhanu Prima Salokha
NPM : 2304220648
Jurusan : D3 Instrumentasi Elektronika dan Industri
Peminatan : Instrumentasi Industri
Tanggal Sidang : 2 Januari 2008
Judul Tugas Akhir:

“PENGENDALI POSISI BERBASIS PC”

Tugas Akhir ini telah diperiksa, diuji, serta disahkan oleh :

PEMBIMBING

(Arief Sudarmaji M.T.)

PENGUJI I

PENGUJI II

PENGUJI III

(Lingga Hermanto, M.si)

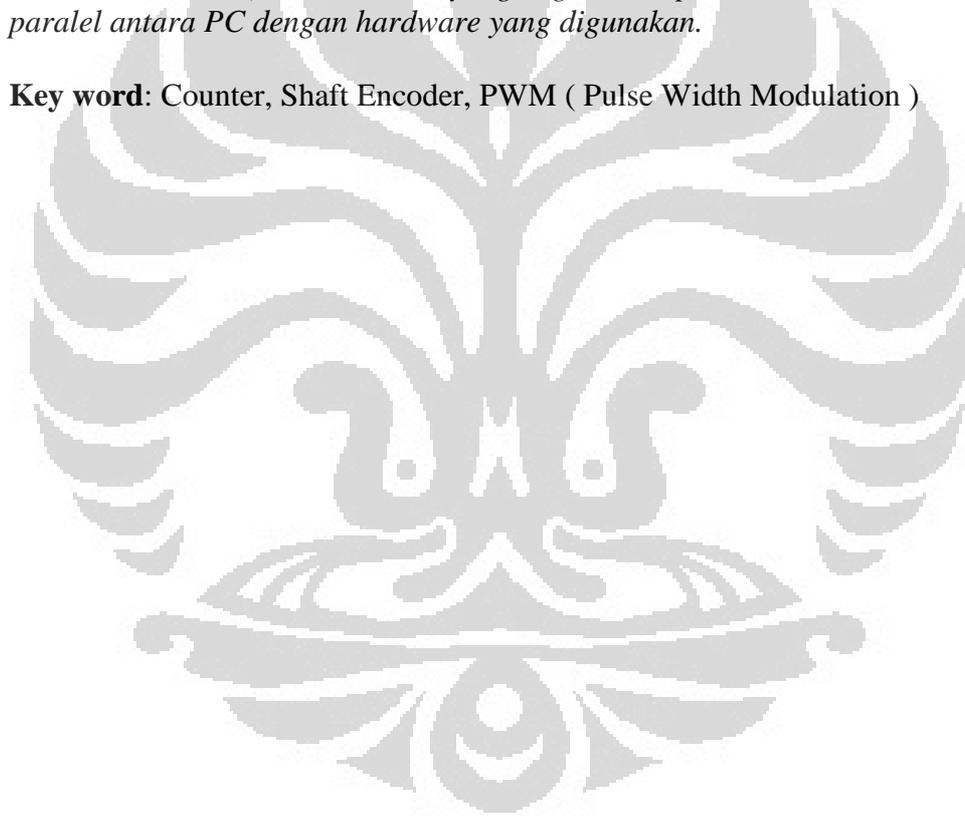
(Dr. Prawito)

(Supriyanto, S.si)

ABSTRAK

Telah dibuat "Rancang Bangun Mekanik Pengendali Posisi" dengan menggunakan suatu penggerak berupa motor dc yang dapat dikendalikan melalui suatu sistem yang terhubung dengan PC. Adapun pengkonversian yang dilakukan dari pergeseran secara linier yaitu dengan menggunakan dua buah batang besi stenleis tanpa drat. Pada motor dc ini terdapat suatu sensor yang berupa shaft encoder yang digunakan sebagai up/down counter (pencacah) dimana pada sensor ini menghasilkan suatu pulsa sebesar 400 dan pada setiap satu putaran motor dc ini menempuh jarak sebesar 10 cm. Adapun range jarak yang digunakan, yaitu pada range 10 cm-95 cm. Pada jarak maksimum, sensor shaft encoder ini dapat menghasilkan cacahan pulsa sebesar 4000. Sedangkan untuk mengatur kecepatan dari motor dc ini digunakan suatu metode PWM (Pulse Width Modulation). Komunikasi yang digunakan pada sistem ini adalah komunikasi paralel antara PC dengan hardware yang digunakan.

Key word: Counter, Shaft Encoder, PWM (Pulse Width Modulation)



KATA PENGANTAR

Puji dan syukur kami panjatkan kepada Tuhan yang Maha Esa, yang telah melimpahkan segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini. Walaupun dalam penyusunan Laporan Tugas Akhir ini penulis menemukan berbagai macam kesulitan, tetapi Tuhan senantiasa memberikan rahmatNya berupa pengetahuan sehingga penulis dapat menyelesaikan laporan tugas akhir berikut.

Penyusunan Laporan Tugas Akhir yang berjudul “PENGENDALI POSISI BERBASISKAN PC” yang bertujuan untuk memenuhi syarat dalam menyelesaikan pendidikan program studi Diploma 3 Instrumentasi Elektronika dan Industri, Departemen Fisika, FMIPA, Universitas Indonesia.

Dalam melaksanakan Tugas Akhir sampai penyelesaian Laporan Tugas Akhir ini, penulis banyak mendapat bantuan dan bimbingan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan terima kasih dan rasa hormat kepada:

1. Tuhan yang Maha Esa, atas segala rahmat dan karunia-Nya yang berlimpah tanpa henti yang telah memberikan secercah pengetahuan serta ilmu yang bermanfaat
2. Kepada kedua orangtuaku yang terkasih, Priyo Utomo dan Rukmi Sri Hadiningsih serta kedua adiku tercinta Yoga Maharditama, Nath Pramesa.
3. Kepada patner kerjaku selama tugas akhir ini, Sistabrani Cidakada, Syamsul Arifin, serta Muhamad Yamidi atas bantuan dan kerja samanya.
4. Drs.Arif Sudarmadji, M.T selaku dosen pembimbing yang telah memberikan petunjuk, kemudahan dalam berpikir dan bimbingan dalam penyelesaian tugas akhir ini.
5. Dr Prawito selaku Ketua program Diploma 3 Instrumentasi Elektronika dan Industri FMIPA, UI.
6. Bapak Surya Dharma, M.Si selaku koordinator Tugas Akhir program Diploma 3 Instrumentasi Elektronika dan Industri FMIPA, UI.
7. Dosen-dosen pengajar Jurusan Instrumentasi Elektronika dan Industri yang selama hampir tiga tahun telah memberikan ilmu-ilmunya.
8. Kepada penghuni kontrakan Cinta, Wahyu, Gunawan, Haeril, Joker, Alet, Aryo.

9. Seluruh rekan-rekan Instrumentasi Industri dan Elektronika angkatan 2004.
10. Buat Riska Nastasha Constantine, atas perhatiannya selama 3 tahun.
11. Tempat ku menghabiskan 5 hari dalam seminggu, kontrakan.
12. Seluruh keluarga besar FMIPA UI.
13. Pak Parno atas bengkelnya, para penjaga gedung fisika malam hari, terima kasih atas tumpangan nginepnya, workshop atas peralatannya, pak Katman atas bornya.
14. Terima kasih atas pengertiannya, perhatiannya, dukungannya, yang tiada hentihentinya selama proses TA ini, Dian Kristiana Suryaningrum.
15. Semua pihak yang secara tidak langsung terlibat dalam pembuatan skripsi ini dan tidak mungkin dapat disebutkan satu persatu, semoga amal baik yang telah dilakukan senantiasa dibalas oleh Tuhan Yang Maha Esa.

Semoga Tuhan yang Maha Esa melimpahkan segala rahmat dan karunia-Nya atas kebaikan Bapak / Ibu dan Saudara/i sekalian.

Semoga penulisan ilmiah ini benar-benar dapat memberikan kontribusi positif dan menimbulkan sikap kritis kepada para pembaca khususnya dan masyarakat pada umumnya untuk senantiasa terus memperoleh wawasan dan ilmu pengetahuan di bidang teknologi.

Menyadari keterbatasan pengalaman dan kemampuan yang dimiliki penulis, sudah tentu terdapat kekurangan serta kemungkinan jauh dari sempurna, untuk itu penulis tidak menutup diri dan mengharapkan adanya saran serta kritik dari berbagai pihak yang sifatnya membangun guna menyempurnakan penyusunan tugas akhir ini.

Akhir kata semoga penyusunan tugas akhir ini dapat memberikan manfaat bagi semua pihak yang bersangkutan, khususnya bagi saya dan umumnya bagi para pembaca.

Depok, 2 Januari 2008

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Penelitian	2
1.3. Batasan Masalah	2
1.4. Deskripsi Singkat	2
1.5. Metode Penulisan	4
1.6. Sistematika Penulisan	5
BAB 2. TEORI DASAR	7
2.1. Motor DC	7
2.1.1. Prinsip Kerja Motor DC	7
2.1.2. Cara Membalik Arah Motor DC.....	11
2.1.3. Cara Mempercepat Putaran Motor DC.....	11
2.2. Shaft Encoder pada Motor DC.....	12
2.3. Metode PWM (<i>Pulse Width Modulation</i>).....	13
2.4. Metode PID (<i>Proportional Integral Differentiator</i>).....	15
BAB 3. PERANCANGAN DAN CARA KERJA SISTEM	17
3.1. Konstruksi Mekanik	17
3.2. Perancangan Hardware	18
3.2.1. Rangkaian Interfacing.....	20

3.2.2. Rangkaian PWM (Pulse Width Modulation).....	21
3.2.3. Rangkaian Penggerak Motor DC	25
3.2.4. Rangkaian Counter	27
3.2.5. Rangkaian Latching	28
3.2.6. Rangkaian Switch	30
3.3. Perancangan SoftWare.....	31
3.3.1. Operasi Start/Stop.....	32
3.3.2. Operasi PID	32
3.3.3. Program Komunikasi Pararel	35
3.3.4. Program Penyimpanan Data	36
BAB 4. Hasil Percobaan dan Perhitungan	37
4.1. Pengujian Pulsa Sensor	37
4.2. Pengujian Perpindahan.....	39
BAB 5. Analisa dan Kesimpulan	41
5.1. Kesimpulan	41
5.2. Saran	42
DAFTAR PUSTAKA	43
LAMPIRAN	

DAFTAR GAMBAR

- Gambar 1. 1. Pengendali Posisi Menggunakan Batang Berulir*
- Gambar 1. 2. Blok Diagram Pengendalian*
- Gambar 2. 1. Kaidah tangan kanan*
- Gambar 2. 2. Susunan Percobaan*
- Gambar 2. 3. Posisi awal gerakan motor*
- Gambar 2. 4. Posisi motor setelah 180⁰*
- Gambar 2. 5. Prinsip Torka*
- Gambar 2. 6. Shaft Encoder pada motor DC*
- Gambar 2. 7. Sinyal PWM dengan duty cycle 50%.*
- Gambar 2. 8. Sinyal PWM dengan duty cycle 10%.*
- Gambar 2. 9. Blok Proses Feedback Control*
- Gambar 2. 10. Proses Feedback Control Pengendali Perpindahan*
- Gambar 2. 11. Blok MV dengan metode MIX*
- Gambar 3. 1. Sistem Kerja Pengendali Perpindahan Berbasis PC*
- Gambar 3. 2. Konstruksi Mekanik Perpindahan*
- Gambar 3. 3. Konfigurasi slot DB-25.*
- Gambar 3. 4. Rangkaian Interfacing*
- Gambar 3. 5. Rangkaian PWM Keseluruhan*
- Gambar 3. 6. Rangkaian Osilator Counter 8 Bit*
- Gambar 3. 7. Rangkaian Komparator*
- Gambar 3. 8. Rangkaian Reset dan Penghasil PWM*
- Gambar 3. 9. Rangkaian CW/CCW*
- Gambar 3. 10. Rangkaian Penggerak Motor DC*
- Gambar 3. 11. Rangkaian Counter*
- Gambar 3. 12. Rangkaian Latching*
- Gambar 3. 13. Rangkaian Switch Motor DC*
- Gambar 3. 14. Flow Chart Program*
- Gambar 3. 15. Operasi Start/Stop*

Gambar 3. 16.(a) Numeric Control pada Front Panel (b) Numeric Control pada Block Diagram

Gambar 3. 17. Operasi dari Persamaan (3.3.)

Gambar 3. 18. Operasi dari Persamaan ($E_n - E_{n-1}$)

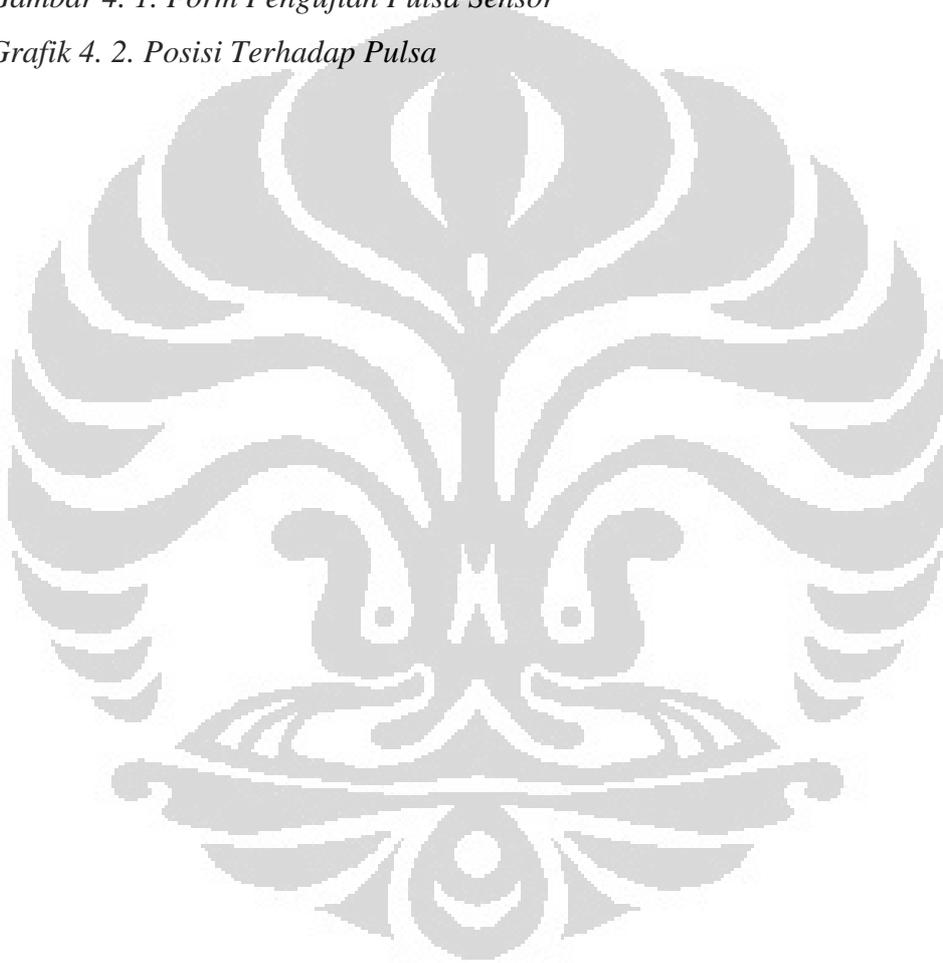
Gambar 3. 19. Operasi add array element

Gambar 3. 20. (a) Outport (b) Inport

Gambar 3. 21. Program Penyimpanan Data

Gambar 4. 1. Form Pengujian Pulsa Sensor

Grafik 4. 2. Posisi Terhadap Pulsa



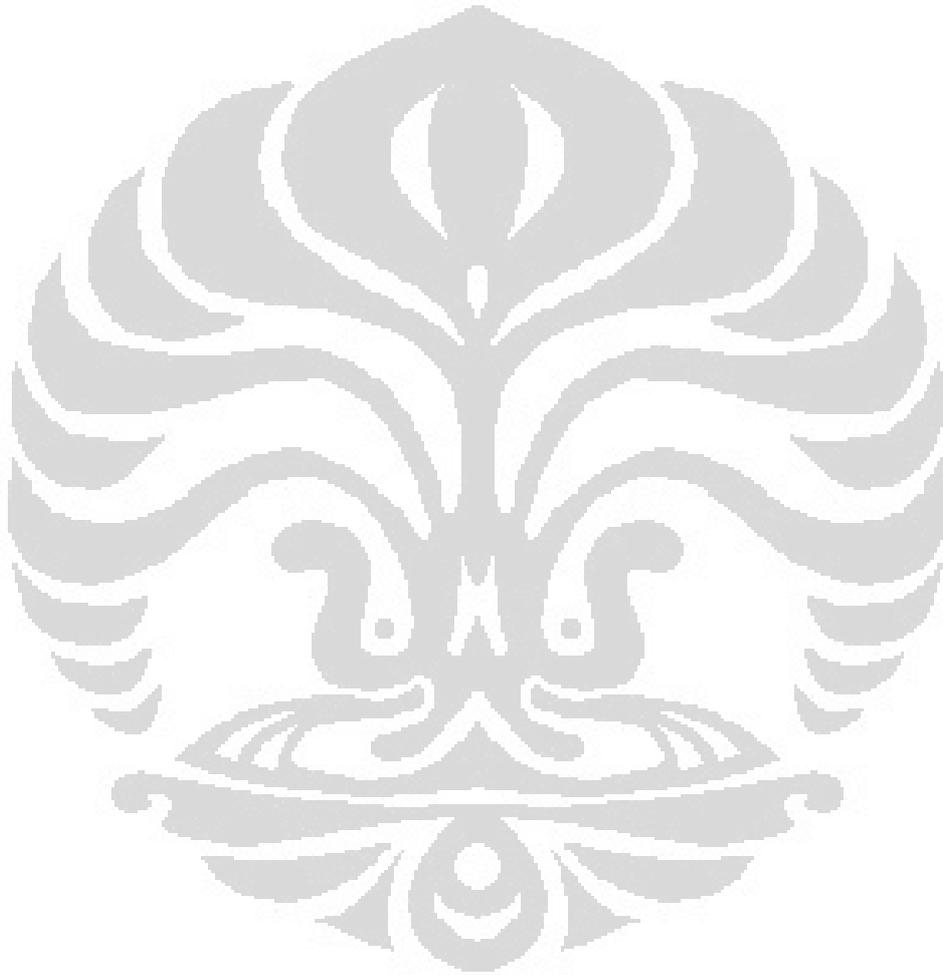
DAFTAR TABEL

Tabel 3. 1. Daftar pin printer port.

Tabel 3. 2. Alamat Port LPT 1.

Tabel 3. 3. Koneksi Port Pararel dengan Hardware

Table 4. 2. Data Perpindahan



BAB 1

PENDAHULUAN

Pada bab ini akan dibahas mengenai latar belakang pembuatan tugas akhir mengenai sistem pengendali posisi berbasis PC yang dapat dikendalikan dan dapat bergerak secara cepat. Serta sistematika penulisan dan bagian-bagian yang terdapat pada laporan ini.

1.1 Latar Belakang

Di era teknologi maju seperti sekarang, dalam bidang industri sangat dibutuhkan suatu alat yang praktis dalam menunjang pekerjaan mereka. Saat ini perusahaan yang bergerak dibidang industri, rata-rata masih menggunakan suatu mekanik yang masih di control secara manual. Untuk itu banyak perusahaan yang mencari jalan keluar untuk menyelesaikan permasalahan tersebut agar proses hasil industrinya dapat berjalan cepat dan efisien.

Perkembangan teknologi saat ini sudah sangat maju dan berkembang. Terutama pada sebuah industri, pengendalian dan pengaturan posisi sangatlah dibutuhkan didalamnya. Dalam bidang industri sekarang ini banyak dibutuhkan suatu alat yang dapat mengatur perpindahan posisi secara cepat agar dapat memudahkan pengerjaan proses industri lebih banyak dan efisien. dimana perkembangan teknologi elektronika sangat berkembang pesat dan maju. Dengan semakin majunya teknologi dan semakin besarnya manfaat yang dapat diberikan, maka semakin banyak para ahli dan pakar teknologi yang terus menerus mengadakan penelitian dan pengembangan yang semata-mata bertujuan untuk memberikan kenyamanan dan kepuasan yang semakin besar untuk manusia itu sendiri.

Salah satu contoh kebutuhan yang dirasa perlu untuk dikembangkan adalah cara menentukan suatu letak atau posisi suatu benda secara tepat dan akurat. Seperti yang kita ketahui bahwa untuk menentukan posisi suatu benda banyak orang yang menggunakan metode manual, namun cara ini kadang masih kurang efektif karena bisa terjadi kesalahan dan keakuratannya tidak dapat dijamin 100%. Oleh sebab itu perlu

adanya perangkat yang dapat mengatasi hal itu. Dengan pengoperasian yang mudah kita dapat memperoleh hasil yang tepat dan akurat. Salah satu contoh kemudahan pengoperasian adalah dengan PC. Dengan pengendali posisi berbasis PC ini diharapkan diperoleh kemudahan-kemudahan pengoperasian suatu perangkat untuk mendapatkan posisi yang akurat.

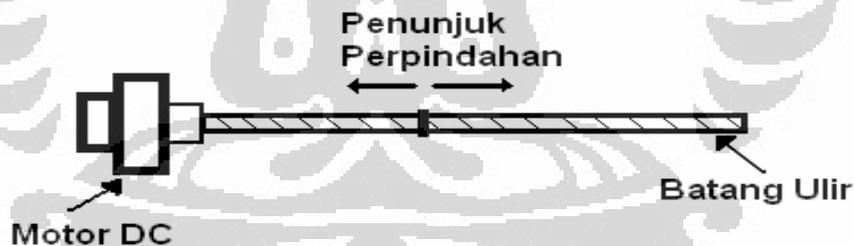
1.2 Tujuan Penelitian

Membuat pengendali posisi yang dapat bergerak cepat dengan menggunakan kendali PC.

1.3 Batasan Masalah

Penelitian ini difokuskan pada pembuatan rancang bangun (mekanik) sistem pengendali posisi dan pembuatan program PC dengan menggunakan software LABVIEW.

1.4 Deskripsi Singkat

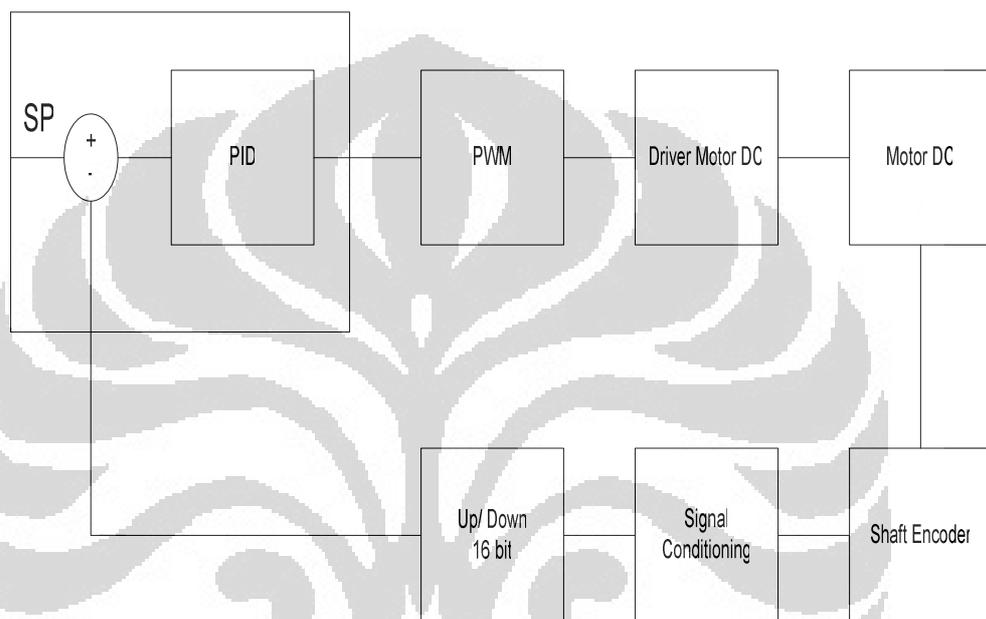


Gambar 1.1. Pengendali Posisi Menggunakan Batang Berulir

Pada tugas akhir sebelumnya yang dibuat oleh mahasiswa Instrumentasi 2004 yang lalu telah di rancang suatu sistem mekanik pengendali posisi dengan menggunakan batang berulir seperti pada gambar 1.1. Untuk proses pengendalian dengan menggunakan mekanik ini sudah bisa dikendalikan dengan baik. Jika memakai sistem

seperti rancangan diatas, benda yang dikendalikan akan bergerak lambat untuk memperoleh set point yang diinginkan.

Oleh karena itu penulis ingin menerapkan suatu pengendali posisi yang bisa dikendalikan dan bergerak dengan cepat. Dengan cara merubah sitem mekanik dengan menggunakan batang stenles yang tidak berulir, serta digerakan oleh motor DC secara langsung. Adapun blok diagram pengendalian yang dibuat adalah sebagai berikut:



Gambar 1.2. Blok Diagram Pengendalian

Untuk dapat merealisasikan suatu perpindahan dengan cepat, maka digunakan cara kerja dari batang besi tidak berulir. Digunakan batang besi yang tidak berulir dikarenakan agar perpindahan posisi dapat berlangsung dengan cepat. Oleh sebab itu digunakan katrol dan motor DC berkecepatan tinggi untuk menggerakkan benda tersebut. Jika benda tersebut berhenti pada posisi yang sudah kita tentukan, maka pegas yang digunakan akan bekerja menahan benda tersebut agar tidak bergeser. Dan dibutuhkan juga tali baja untuk semakin menguatkan benda tersebut supaya tidak bergeser dan berada tepat pada posisi yang sudah ditentukan. Benda tersebut dapat di atur posisinya pada keadaan semula, dengan mengendalikan motor DC tersebut agar dapat memutar berlawanan arah, sehingga motor DC tersebut menggerakkan katrol sehingga benda dapat kembali pada keadaan semula.

1.5 Metode Penulisan

Metode penelitian dalam pembuatan Pengendali Posisi berbasis PC adalah sebagai berikut :

1 Studi Literatur

Penulis menggunakan metode ini untuk memperoleh informasi yang berkaitan dengan penelitian yang dilakukan dengan mengacu kepada buku-buku, *data sheet*, informasi dari internet, dan lain-lain.

2 Perancangan Sistem

Pada tahap ini penulis berusaha untuk membuat suatu rancangan sistem kontrol baru yang hendak dibuat di dalam penelitian nantinya. Di sini saya akan mempelajari sistem kontrol yang telah ada dan juga menganalisa kembali kelebihan dan juga kekurangan dari sistem kontrol tersebut agar dapat memperoleh sebuah rancangan sistem kontrol baru yang lebih baik.

3 Pembuatan Alat

Langkah selanjutnya merupakan pembuatan alat dari rancangan yang telah diperoleh pada tahap sebelumnya, sesuai dengan tujuan yang diinginkan.

4 Pengujian Sistem dan Komponen Pendukung

Hal pertama yang harus diuji terlebih dahulu adalah sistem alat (hardware), kemudia baru pemrograman (software), kemudian komunikasi antara hardware dan software.

5 Pengumpulan Data

Di sini alat diuji secara keseluruhan sebagai suatu kesatuan sistem. Setelah itu dapat dilihat apakah perangkat keras dan lunak sudah dapat bekerja dengan benar

ataukah masih dibutuhkan beberapa perbaikan. Jika alat sudah dapat bekerja dengan benar, maka dapat dilakukan pengumpulan data yang dianggap penting atau diinginkan.

6 Penulisan Hasil Penelitian

Hasil dari pengujian dan pengumpulan data kemudian dianalisa. Dari sini kita dapat menarik kesimpulan dari penelitian yang telah kita lakukan. Beberapa saran juga dapat kita ajukan sebagai bahan perbaikan untuk penelitian lebih lanjut.

1.6 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari beberapa bab. Agar lebih mempermudah pemahaman dan pembacaan, maka laporan tugas akhir ini disusun menjadi seperti di bawah ini :

BAB I. PENDAHULUAN

Pada bab ini berisi tentang latar belakang, tujuan penelitian pembatasan masalah, metode penelitian dan sistematika penulisan.

BAB II. TEORI DASAR

Pada Bab ini berisi tentang konsep yang mendasari cara kerja dari sistem Pengendali Posisi Berbasis PC yang terkoneksi dengan hardware.

BAB III. PERANCANGAN SISTEM

Merupakan penjelasan pembuatan rancangan sistem kontrol baik hardware atau software dengan menginputkan program atau data yang akan digunakan sebagai pengujian dari alat tersebut.

BAB IV. PENGUJIAN SISTEM DAN PENGAMBILAN DATA

Sistem yang telah dirancang kemudian diuji dengan parameter-parameter yang terkait. Pengujian ini meliputi pengujian *software* dan *hardware* dilakukan secara

simultan. Di samping pengujian, proses pengambilan data kerja sistem ini juga dituliskan di bab ini untuk memastikan kemampuan sistem secara keseluruhan. Dari hasil ini dapat dilakukan analisa terhadap kerja sistem, sehingga dapat diketahui apa yang menjadi penyebab dari kendala atau kegagalan bila selama kegiatan penelitian ditemui hal-hal tersebut.

BAB V. KESIMPULAN

Pada Bab ini berisi tentang kesimpulan dari hasil penelitian yang telah dilakukan yang berkaitan dengan Pengendali Posisi Berbasis PC.



BAB 2 TEORI DASAR

Dalam bab ini akan dibahas mengenai teori yang mendasari dilakukannya penelitian ini. Adapun teori-teori tersebut antara lain:

2.1. Motor DC

Salah satu komponen yang diperlukan dalam system pengendali adalah *actuator*. *Actuator* adalah komponen pertama untuk melakukan gerakan, mengubah energi elektrik menjadi gerakan mekanik. Adapun jenis actuator salah satunya adalah motor listrik. Motor listrik dikelompokkan menjadi motor DC dan motor AC, perbedaannya terdapat pada tegangan yang menggerakkannya. Motor AC digerakkan oleh tegangan bolak-balik (AC), sedangkan motor DC digerakkan oleh tegangan searah (DC)

2.1.1. Prinsip Kerja Motor DC

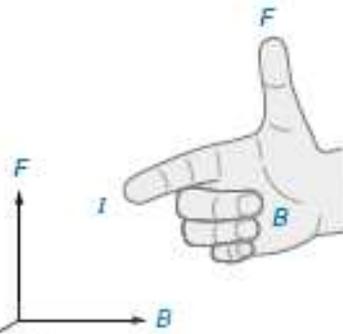
Prinsip kerja motor DC dapat dijelaskan dengan teori *elektromagnetik*. Misal sebuah kawat berarus yang dipengaruhi medan magnet luar akan mengalami gaya yang disebut gaya magnet yang besarnya ditunjukkan pada persamaan:

$$F = B \cdot I \cdot L \sin \alpha \dots\dots\dots(2.1)^{[1]}$$

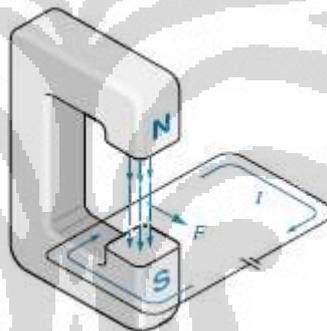
Dimana:

- F** = gaya magnet (Newton)
- B** = medan magnet luar (Wb/m²)
- I** = kuat arus (Ampere)
- L** = panjang kawat (Meter)
- α** = sudut yang dibentuk medan magnetic dengan arus

Pada sebuah kawat berarus listrik di dalam pengaruh medan magnet, maka arah gaya **F** dapat ditentukan dengan kaidah tangan kanan:



Gambar 2.1. Kaidah tangan kanan^[8]

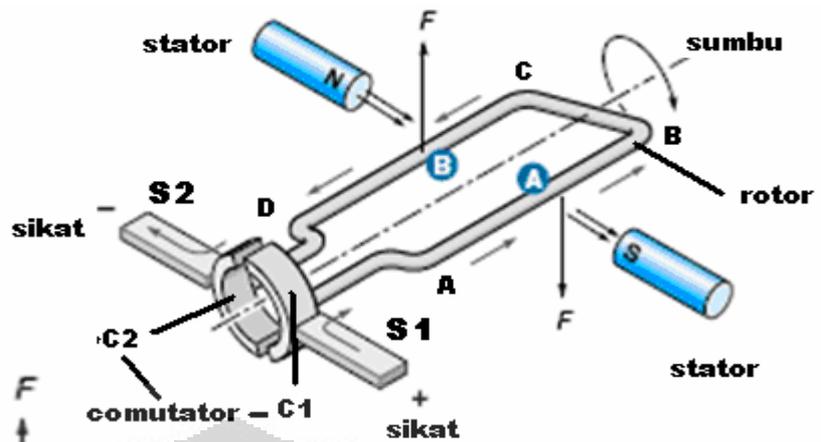


Gambar 2.2. Susunan Percobaan^[8]

Motor *dc* terdiri dari bagian-bagian yang dapat menggerakkan motor tersebut, yaitu:

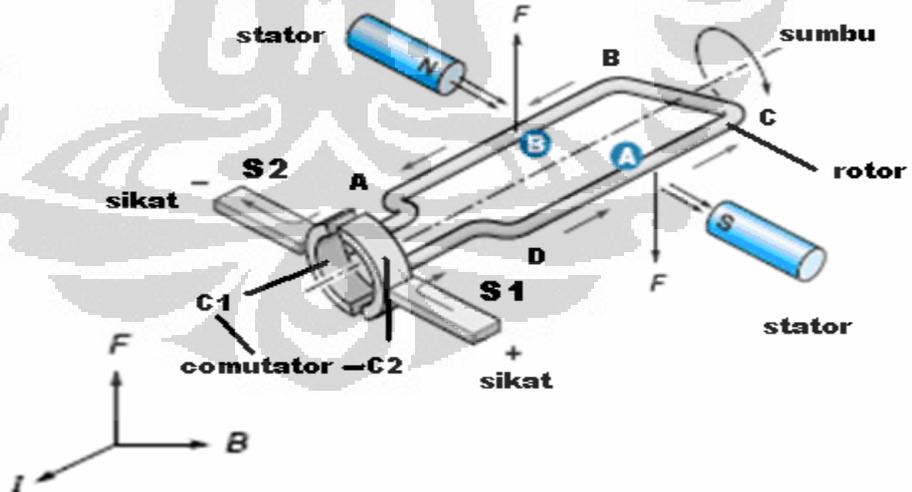
1. **Rotor**, yaitu bagian yang berputar pada motor berupa kumparan kawat.
2. **Stator**, yaitu bagian yang diam pada motor berupa magnet.
3. **Komutator**, yaitu cincin belah yang berfungsi sebagai penukar arus.
4. **Sikat**, yaitu sepasang batang grafit yang menempel pada komutator tetapi tidak berputar.

Cara kerja motor *dc* dapat dilihat pada gambar 2.2 berikut:



Gambar 2.3. Posisi awal gerakan motor^[8]

Misalkan posisi mula-mula seperti pada gambar 2.3 arus listrik mengalir dari kutub (+) baterai melalui sikat S1 – cincin C1- rotor ABCD – cincin C2 – sikat S2 – kembali ke kutub (-) baterai. Ketika rotor CD yang dekat dengan kutub utara mengalami gaya ke atas dan sisi rotor AB yang dekat dengan kutub selatan mengalami gaya ke bawah. Akibatnya rotor ABCD berputar searah jarum jam.



Gambar 2.4. Posisi motor setelah 180⁰^[8]

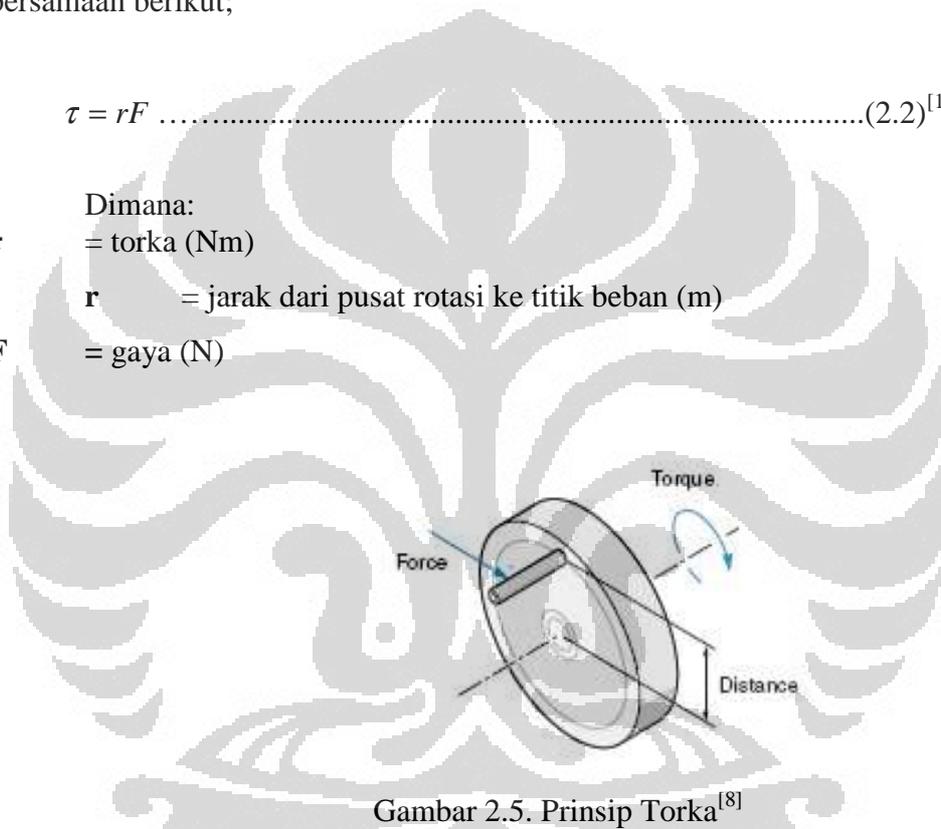
Setelah setengah putaran (180⁰), terjadi pertukaran posisi antara sikat dan comutator. Sekarang, C2 menyentuh sikat S1 dan C1 menyentuh sikat S2. Sehingga

arus mengalir dari kutub (+) baterai menuju kutub (-) melalui sikat 1 (S1), Comutator 2 (C2), Rotor DCBA, Comutator 2 (C2), dan sikat 2 (S2). Pertukaran posisi antara sikat dan comutator mengakibatkan motor terus berputar

Selama motor berputar menghasilkan torka ($\tau = \text{Torque}$). Torka merupakan analogi gaya dari gerak translasi untuk gerak rotasi. Karena torka ini dihasilkan oleh sistem elektromagnet, maka disebut torka elektromagnet (electromagnetic torque). Torka yang dihasilkan motor ini mempunyai nilai yang besarnya ditunjukkan pada persamaan berikut;

$$\tau = rF \dots\dots\dots(2.2)^{[1]}$$

- Dimana:
- τ = torka (Nm)
 - r = jarak dari pusat rotasi ke titik beban (m)
 - F = gaya (N)



Gambar 2.5. Prinsip Torka^[8]

Sedangkan ketika terjadi putaran persamaan torka menjadi:

$$\tau = I\alpha \dots\dots\dots(2.3)^{[1]}$$

- Dimana:
- τ = torka putaran
 - I = momen inersia
 - α = kecepatan sudut (rad/s)

2.1.2. Cara Membalik Arah Motor DC

Arah gerakan motor arus searah dapat diatur dengan dua cara yaitu mengubah polarisasi arah arus searah pada belitan medan magnet (+) dan (-), atau dengan mengubah arah arus dengan menukar (+) dan (-) pada sikat.

Pada prinsipnya membalik arah motor searah memang dengan dua cara yang telah disebutkan di atas, namun dalam suatu rangkaian elektronika kita memerlukan suatu rangkaian penggerak motor yang dapat membalik arah gerak motor dengan mudah misalnya dengan menggunakan transistor. Transistor pada rangkaian pembalik putaran motor berfungsi sebagai saklar (*switching*).

2.1.3. Cara Mempercepat Putaran Motor DC

Kecepatan putaran motor dc dapat ditingkatkan dengan memperbesar tegangan yang masuk ke motor, sehingga dapat mengakibatkan arus yang masuk ke motor menjadi besar pula. Hal ini sesuai dengan hukum Kirchoff berikut ini:

$$V = I \times R \dots\dots\dots(2.4)^{[1]}$$

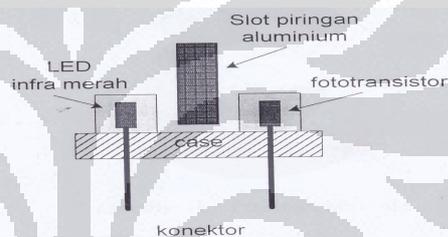
Dimana:

- V** = Tegangan (Volt)
- I** = Besar arus (Ampere)
- R** = Hambatan (Ohm)

Dengan hambatan yang tetap dan tegangan diperbesar akan mengakibatkan arus menjadi besar pula. Dengan arus yang diperbesar maka akan menyebabkan gaya (**F**) menjadi besar pula sesuai dengan persamaan 1.1 di atas. Dan apabila **F** semakin besar maka kekuatan rotor akan semakin besar dan berdampak pada makin cepatnya putaran motor. Dari persamaan 1.1 dapat diambil kesimpulan bahwa semua unsur yang mempengaruhi nilai **F** dapat mempercepat putaran motor, yaitu dengan memperpanjang lilitan (memperbesar **I**), dan memperbesar medan magnet (**B**).

2.2. Shaft Encoder pada Motor DC

Bagian ini adalah bagian yang penting dari motor DC, sebab ketika motor DC tersebut berputar, ia menghasilkan pulsa, dan pulsa tersebut dapat dibaca oleh program karena adanya Shaft Encoder tersebut. *Shaft encoder* merupakan suatu sensor yang digunakan untuk menghitung berapa banyak motor melakukan pengcounter dalam satu putaran atau dengan kata lain mengubah putaran mekanis menjadi data digital. Pengcounteran yang dimaksudkan adalah berasal dari piringan yang ada pada motor. *Shaft encoder* ini pada umumnya digunakan untuk menghitung berapa banyak putaran dalam persekian menit (Rpm).



Gambar 2.6. Shaft Encoder pada motor DC

Seperti pada gambar di atas slot piringan aluminium terhubung dengan lengan motor sehingga ketika motor berputar maka slot piringan tersebut akan berputar pula. Pada *shaft encoder* ini terdapat bagian pengirim cahaya (*Transmitter*) dan bagian penerima cahaya (*Receiver*). Antara bagian pengirim cahaya dan bagian penerima cahaya diberi piringan untuk mengkodekan data. Piringan pengkode data ini ditempelkan pada motor dc . Apabila motor ini berputar, maka piringan pengkode data juga ikut berputar. Sedangkan sensor berada dalam kedudukan yang tetap. Prinsip dari sensor ini akan memberi output low (0) jika antara *transmitter* dan *receiver* terhalang piringan pengkode data. Sebaliknya bila antara transmitter dan receiver tidak terhalang piringan pengkode data maka output yang dihasilkan adalah high (1).

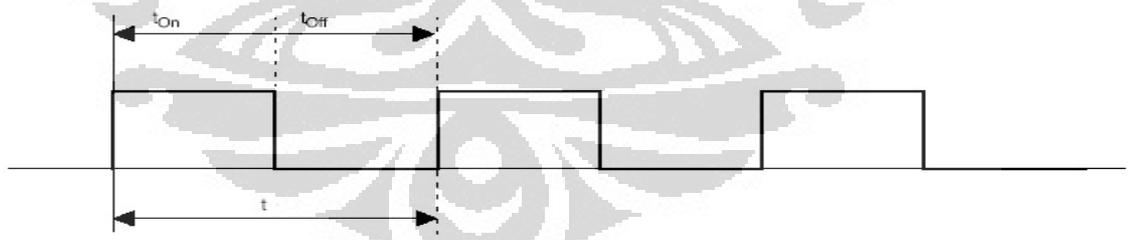
Dari sensor *shaft encoder* didapatkan data-data berbentuk pulsa dan selanjutnya pulsa-pulsa tersebut diolah menjadi *up-down counter*. *Up-down counter* disini

menandakan adanya pergeseran posisi dari penunjuk perpindahan dari konstruksi mekanik. Jika penunjuk bergerak maju, maka akan mencacah naik (*up counter*). Sedangkan apabila penunjuk bergerak mundur rangkaian akan mencacah turun (*down counter*).

2.3. Metode PWM (*Pulse Width Modulation*)

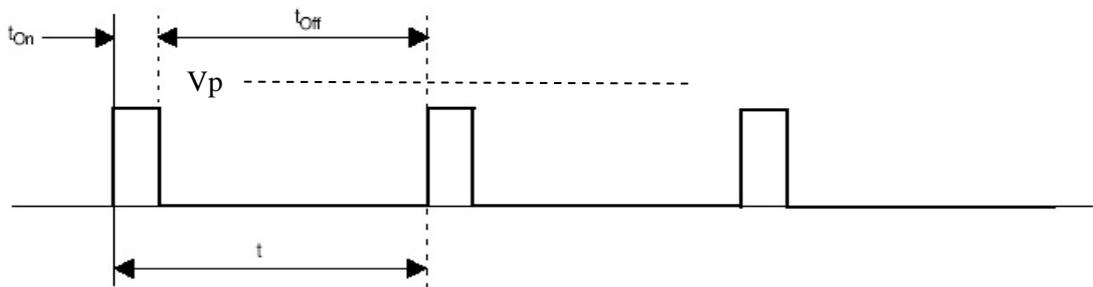
Rangkaian PWM termasuk dalam rancangan hardware dari PC ke konstruksi mekanik. PWM adalah suatu teknik yang digunakan untuk mengontrol kerja dari suatu alat atau menghasilkan suatu tegangan DC yang variabel ^[4]. Rangkaian PWM adalah rangkaian yang lebar pulsa tegangan keluarannya dapat diatur atau dimodulasi oleh sebuah sinyal tegangan modulasi. Disamping itu kita dapat menghasilkan suatu sinyal PWM dengan menentukan frekuensi dan waktu dari variabel ON dan OFF. Pemodulasian sinyal yang beragam dapat menghasilkan *duty cycle* yang diinginkan. Gambar 2.7 memperlihatkan sinyal kotak dengan *duty cycle* 50%. *Duty cycle* adalah rasio dari waktu ON (t_{on}) terhadap periode total dari sinyal ($t = t_{on} + t_{off}$). Dengan persamaan (2.5) :

$$D = \frac{t_1}{t_1 + t_2} \dots\dots\dots(2.5)$$



Gambar 2.7. Sinyal PWM dengan *duty cycle* 50%.

Keadaan *duty cycle* yang bermacam-macam, rata-rata output dari tegangan DC dapat dikontrol. Seperti pada gambar 3.5 memperlihatkan sinyal kotak dengan *duty cycle* 10%.



Gambar 2.8. Sinyal PWM dengan *duty cycle* 10%.

Hubungan antara pulsa hingga bisa mengatur kecepatan dari motor dc, dapat kita lihat pada pulsa diatas. Waktu (t_{on}) dapat dijabarkan dengan menggunakan rumus seperti dibawah ini:

$$V_{rms} = \sqrt{\int_0^{t_{on}} V_p^2 dt + \int_{t_{on}}^0 0 dt} \dots\dots\dots (2.6)$$

Maka persamaan diatas disederhanakan menjadi:

$$= V_p \cdot t_{on} / T \dots\dots\dots (2.7)$$

Hubungan dengan daya pada motor dc adalah sebagai berikut:

$$P = V^2 / R = (V_p \cdot t_{on} / T)^2 / R = V_p^2 / R * t_{on} / T = V_p^2 / RT * t_{on}$$

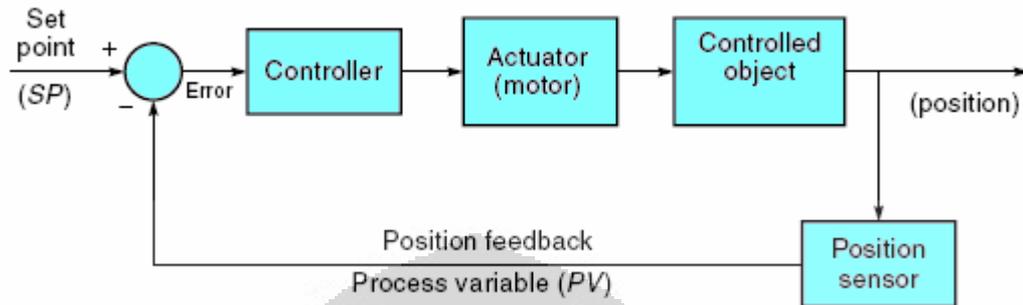
Dari persamaan yang telah diturunkan tersebut didapat bahwa:

$$T_{on} = \text{Angka PWM}$$

2.4. Metode PID (Proportional Integral Differentiator)

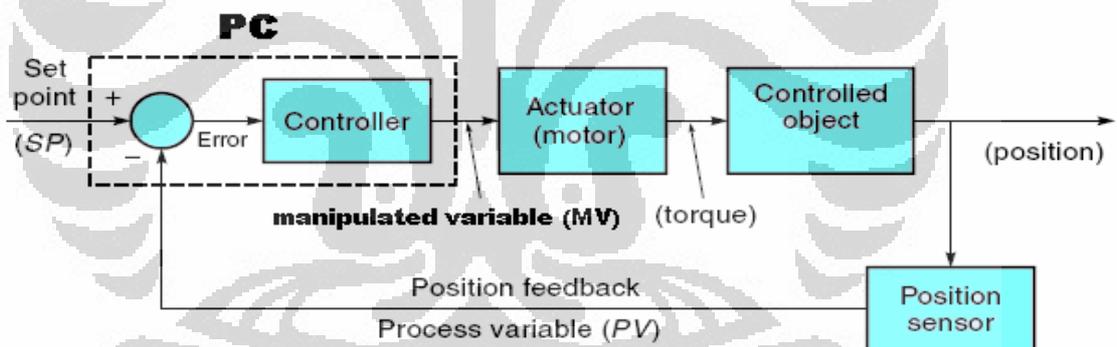
Pada bagian ini dibahas mengenai program yang digunakan untuk merealisasikan pengendalian dengan metode PID (*Proportional Integral Derrivative*). Sedangkan PID sendiri merupakan salah satu bentuk proses *closed-loop control* atau *feedback control*. Adapun inti dari *feedback control* adalah adanya *position feedback* atau biasa disebut

Process Variable (PV). Sedangkan blok proses dari *feedback control* adalah sebagai berikut:



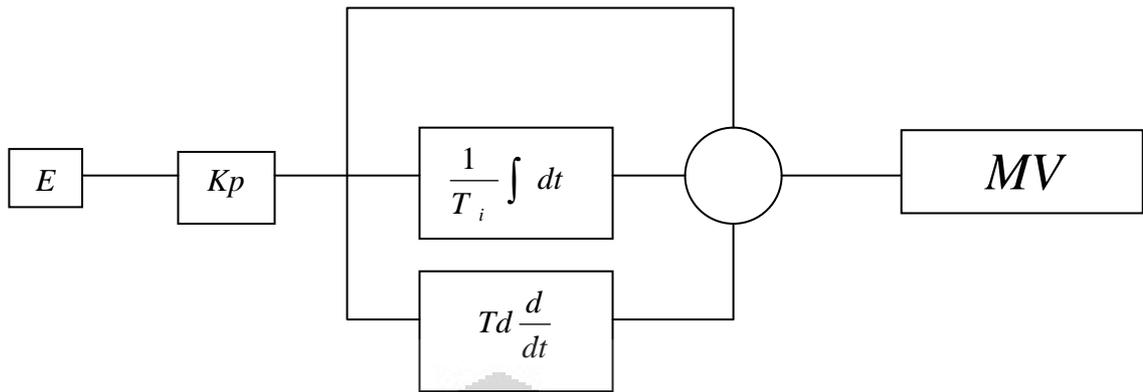
Gambar 2.9. Blok Proses *Feedback Control*

Sedangkan proses dari *PID control* yang dilakukan dalam penelitian ini dapat dilihat pada gambar 2.10. Adapun *software* yang digunakan adalah untuk mendapatkan nilai *MV (manipulated variable)*.



Gambar 2.10. Proses *Feedback Control* Pengendali Perpindahan

Sedangkan *MV (Manipulated Variable)* adalah proses untuk mengatur variabel dan metode yang digunakan adalah dengan metode *Mix*. Metode ini memiliki blok skema sebagai berikut:



Gambar 2.11. Blok MV dengan metode MIX

Adapun persamaan yang didapat dari blok diagram di atas adalah:

$$MV = KpE(t) + \frac{Kp}{Ti} \int E(t)dt + KpTd \frac{dE(t)}{dt} \dots\dots\dots(2.8)$$

Persamaan di atas merupakan persamaan dengan dari *Analog PID Controller*. Sedangkan penelitian yang dilakukan mengacu pada cara kerja dari *Digital PID Controller*. Sehingga persamaan di atas berubah menjadi:

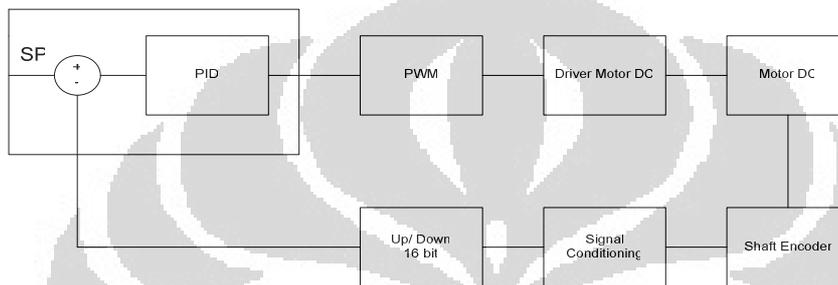
$$MV = KpE + \frac{KpTS}{Ti} \sum E + \frac{KpTd}{TS} (E_n - E_{n-1}) \dots\dots\dots(2.9)$$

Adapun perbedaan dari keduanya adalah pada fungsi waktu. Untuk proses analog menggunakan *kontinuitas* waktu, sedangkan untuk proses digital tidak mengnut adanya *kontinuitas* waktu. Fungsi tersebut diubah dalam bentuk *Time Sampling (TS)* yaitu penghitungan waktu secara periodik. .

BAB 3

PERANCANGAN DAN CARA KERJA SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem beserta cara kerja dari masing-masing konstruksi mekanik, *hardware* serta *software* yang digunakan penulis dalam penyusunan alat ‘Pengendalian Perpindahan Berbasis PC’.

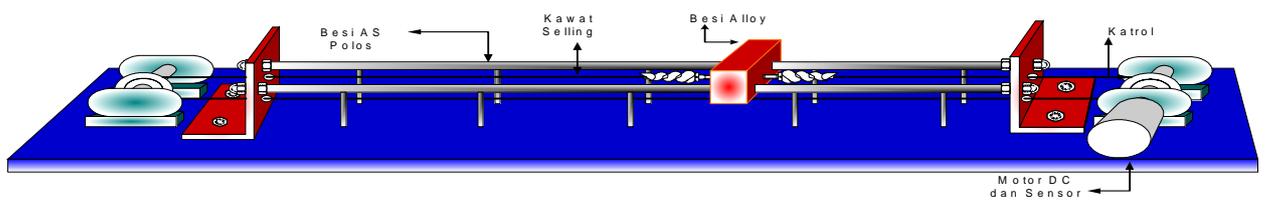


Gambar 3.1. Sistem Kerja Pengendali Perpindahan Berbasis PC

Dari gambar di atas terlihat bahwa pada keseluruhan sistem terbagi dalam tiga bagian yaitu konstruksi mekanik, *hardware*, serta *software*.

3.1. Konstruksi Mekanik

Untuk dapat merealisasikan suatu perpindahan, digunakan cara kerja dari batang besi stensis atau besi panjang yang lurus dengan menggunakan penyangga agar besi stensis tersebut tidak mengalami kebengkokan. Apabila motor tersebut berputar maka motor tersebut akan menggerakkan tali baja yang di pasang pada tengah-tengah sisi kotak alloy, dan kotak alloy tersebut dapat berpindah dari kanan ke kiri mengikuti lajur pada besi stensis tersebut. Adapun untuk dapat menggerakkan kotak besi tersebut secara otomatis digunakan sebuah motor DC yang terhubung dengan tali baja yang dikaitkan pada polly block, serta tali baja tersebut dikaitkan pada pegas. Pegas tersebut digunakan agar bisa meredam pergerakan dari gaya tarik dari tali besi tersebut, dan juga untuk melindungi drat dari besi alloy agar tidak terkena tarikan secara langsung yang dapat mengakibatkan rusaknya drat tersebut.



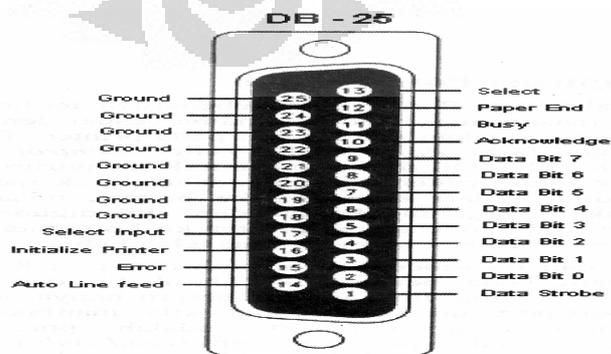
Gambar 3.2. Konstruksi Mekanik Perpindahan

Adapun batang besi stenlis yang digunakan memiliki panjang 1.5 m dengan diameter 12 mm yang digunakan sebagai rel untuk pergeseran dari kotak besi tersebut. Tali baja yang dikaitkan pada per digunakan untuk menggerakkan kotak tersebut, dan kekencangan dari tali besi tersebut dapat diatur dengan memutar baut pada kotak besi tersebut, tali yang digunakan yaitu sepanjang 3 m. Untuk memperlancar pergerakan motor tali digunakan *polly block*.

Motor DC yang digunakan dalam penyusunan konstruksi mekanik ini dapat menghasilkan 400 pulsa cacahan. Semakin banyak pulsa cacahan yang dihasilkan motor, maka perpindahan posisi yang diinginkan akan lebih *presisi*.

3.2. Perancangan Hardware

Pada bagian hardware terbagi menjadi dua bagian yaitu dari PC ke mekanik dan dari mekanik ke PC. Adapun proses penerimaan dan pengiriman data ini melalui *port paralel* (LPT) yang mempunyai alamat 0x378 H atau 888. Dimana pada komputer menggunakan slot DB-25 *female* yang terdapat di belakang komputer. Adapun konfigurasi slot DB-25 *female* adalah sebagai berikut :



Gambar 3.3. Konfigurasi slot DB-25.

Port LPT 1 merupakan salah satu *port* yang dimiliki oleh komputer. Biasanya *port* ini dipakai untuk menghubungkan PC dengan *printer*. Keunggulan dari *port* ini terletak pada kemampuannya untuk menyampaikan data lebih cepat, karena *port* ini menggunakan komunikasi secara paralel. Meskipun memerlukan banyak kabel, namun memilih *port* ini sebagai *interface* sangat tepat karena jarak alat tidak terlalu jauh dari komputer. Fungsi dari masing-masing Pin dalam DB 25 dapat dilihat pada tabel 3.1 berikut ini :

Tabel 3.1. Daftar pin *printer port*.

Pin DB-25	Nama Pin	Keterangan	Arah input/output	Sifat
		<i>Printer Control 0</i> (PC-0)		
		<i>Data Port</i> (DP0 - DP9)		
1	<i>Strobe</i>	<i>Printer Status 6</i> (PS-6)	<i>Output</i>	<i>Inverting</i>
2 – 9	<i>Data Output</i>	<i>Printer Status 7</i>	<i>Output</i>	<i>Normal</i>
10	<i>Acknowledge</i>	(PS-7)	<i>Input</i>	<i>Normal</i>
11	<i>Busy</i>	<i>Printer Status 5</i>	<i>Input</i>	<i>Inverting</i>
12	<i>Paper End</i>	(PS-5)	<i>Input</i>	<i>Normal</i>
13	<i>Select</i>	<i>Printer Status 4</i>	<i>Input</i>	<i>Normal</i>
14	<i>Autofeed</i>	(PS-4)	<i>Output</i>	<i>Inverting</i>
15	<i>Error</i>	<i>Printer Control 1</i>	<i>Input</i>	<i>Normal</i>
16	<i>Init</i>	(PC-1)	<i>Output</i>	<i>Normal</i>
17	<i>Select IN</i>	<i>Printer Status 3</i>	<i>Output</i>	<i>Inverting</i>
18-25	<i>Ground</i>	(PS-3)		
		<i>Printer Control 2</i> (PC-2)		
		<i>Printer Control 3</i> (PC-3)		

Port LPT 1 akan dapat diakses apabila alamat *port* tersebut pada komputer diketahui. Cara mengakses *port* ini tentunya dengan *software*. Adapun alamat dari LPT 1 pada PC sebagaimana tercantum dalam tabel 3.2.

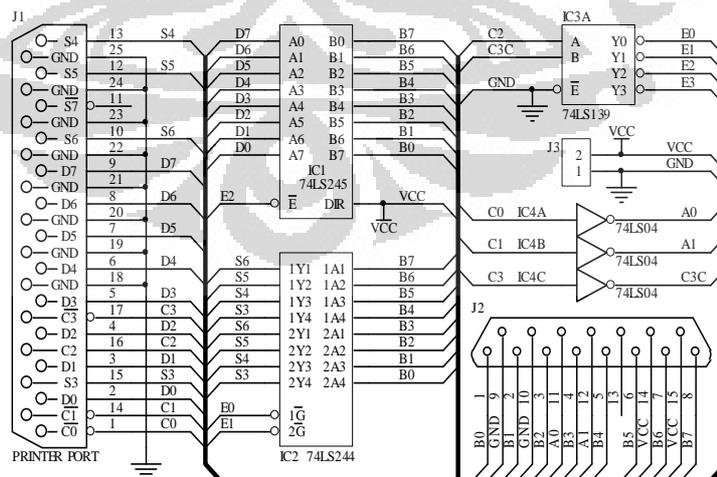
Tabel 3.2. Alamat Port LPT 1.

Nama	Alamat
Data Port (DP)	378H (888)
Printer Status (PS)	379H (889)
Printer Control (PC)	37AH (890)

Umumnya LPT 1 pada PC digunakan untuk mengirimkan data dari PC ke *printer*. Hal ini berarti bahwa LPT 1 hanya dapat dipakai untuk *output* data saja. Oleh karena itu, diperlukan teknik tertentu untuk menjadikan LPT 1 sebagai *input* dan *output* data.

3.2.1. Rangkaian Interfacing

Rangkaian *interfacing* ini diperlukan untuk komunikasi data dengan hardware yang digunakan (mengirim serta menerima data). Adapun rangkaian interfacing yang digunakan adalah sebagai berikut:



Gambar 3.4. Rangkaian *Interfacing*

Rangkaian *interfacing* disini berfungsi untuk menyatukan dua *protocol* yang berbeda, antara PC dengan *hardware* yang digunakan. Adapun jalur komunikasi yang digunakan adalah port paralel. Seperti pada table 3.1, bahwa port printer memiliki 8 pin yang berfungsi sebagai *data output* dan 4 pin yang berfungsi sebagai *input* yang bersifat normal. Normal disini mengandung pengertian bahwa kondisi data yang masuk ke port printer sama dengan kondisi data yang dikeluarkan oleh hardware. Berbeda dengan sifat *inverting*, yang mengandung pengertian bahwa kondisi yang masuk ke port printer berkebalikan dengan kondisi yang dikeluarkan oleh *hardware*.

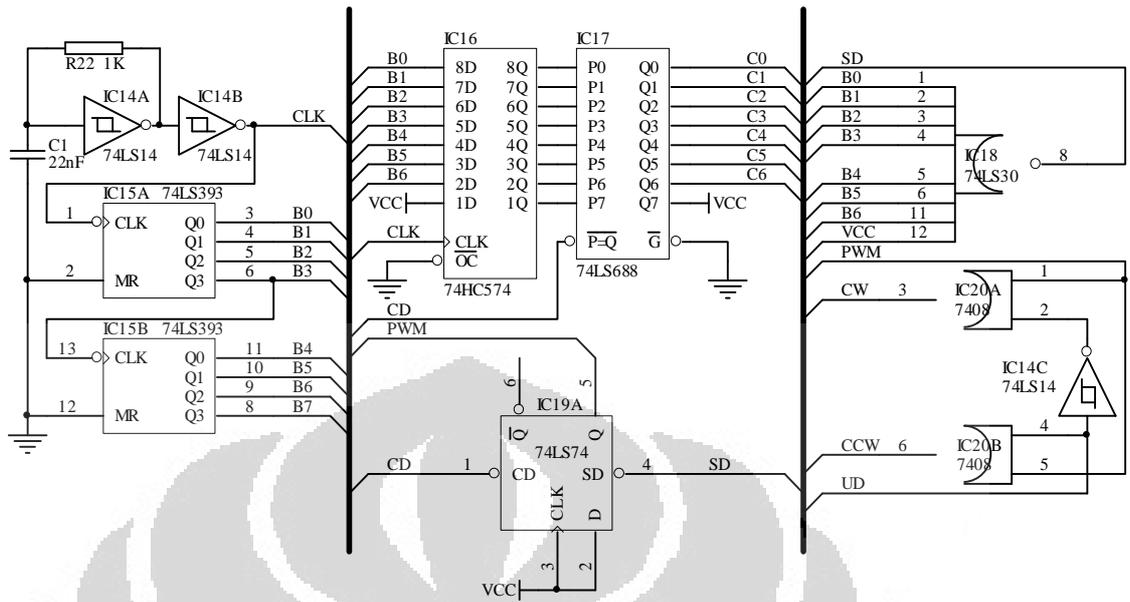
Dalam pembuatan rangkaian *interfacing* ini penyusun memanfaatkan 8 jalur *data output* dan 4 jalur *input* normal tersebut sebagai jalur pengiriman dan penerimaan data dari hardware yang digunakan.

Untuk jalur pengiriman data, digunakan IC 74LS245. Seperti yang terlihat pada gambar skematik di atas bahwa lajur D (D0 hingga D7) terhubung dengan port paralel sedangkan lajur B (B0 hingga B7) terhubung dengan *hardware* luar. Adapun jalur DIR terhubung dengan VCC. Hal ini dimaksudkan agar data berjalan dari lajur B menuju lajur A. Sedangkan jalur *enable* (E) dapat dikendalikan dari PC. Jalur *enable* ini berfungsi sebagai penentu kapan data dari lajur B dialirkan menuju lajur A. Jika kondisinya *low* maka data akan dialirkan, dan jika kondisinya *high* data tidak dialirkan.

Adapun jalur penerimaan data, digunakan IC 74LS244. IC ini dapat membagi 8 bit data menjadi 4 bit data yang dikirim secara bertahap. Adapun pin E0 (*enable 0*) yang berfungsi mengalirkan 4 bit data yang pertama dapat dikontrol dari PC. Begitu juga dengan E1 (*enable 1*) yang berfungsi mengalirkan 4 bit data berikutnya juga dapat dikontrol dari PC. Sedangkan semua jenis data baik data pengiriman maupun data penerimaan terhubung dengan *hardware* luar melalui DB 15.

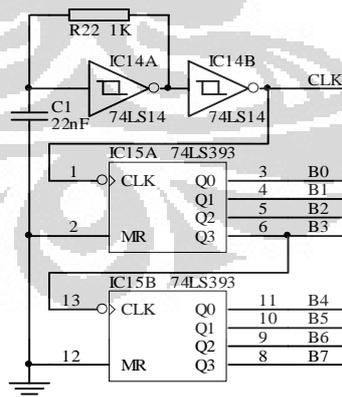
3.2.2. Rangkaian PWM (*Pulse width Modulation*)

Metode pengontrol *output* ini dapat diaplikasikan dalam banyak hal. Salah satunya adalah pengontrol kecepatan motor DC. Adapun rangkaian PWM yang digunakan sebagai berikut:



Gambar 3.5. Rangkaian PWM Keseluruhan

Dengan rangkaian PWM di atas pulsa yang dihasilkan berbentuk pulsa digital. Untuk menghasilkan suatu pulsa dibutuhkan rangkaian osilator (penghasil pulsa). Adapun rangkaian yang digunakan dapat menghasilkan pulsa 8 bit seperti pada gambar di bawah:



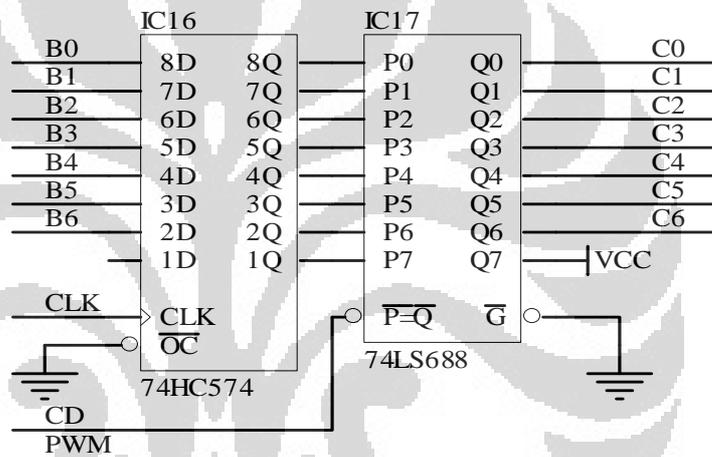
Gambar 3.6. Rangkaian Osilator Counter 8 Bit

Dari output 74LS14 dihasilkan pulsa yang terus berdetak. Adapun frekuensinya dipengaruhi dari nilai R dan C yang digunakan sesuai dengan persamaan:

$$f = \frac{1}{2\pi RC} \dots\dots\dots(3.1)$$

Selanjutnya output dari 74LS14 masuk ke input 74LS393. Dari sini dihasilkan pulsa counter 8 bit.

Pada rangkaian di atas pulsa akan terus berdetak tanpa bisa diatur dari PC. Untuk dapat diatur dari PC diperlukan rangkaian Komparator seperti pada gambar berikut:

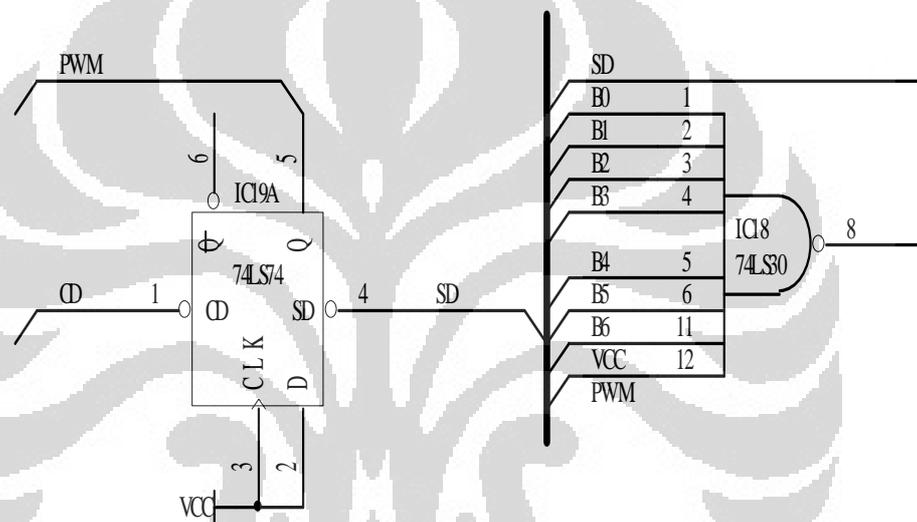


Gambar 3.7. Rangkaian Komparator

Rangkaian ini berfungsi sebagai pembanding antara data dari PC dengan pulsa yang terus berdetak dari rangkaian osilator^[6]. Jika data dari PC dan pulsa dari rangkaian osilator telah sama, maka data tersebut nantinya yang akan diolah menjadi data PWM. Rangkaian Komparator ini menggunakan dua buah IC yaitu 74HC574 dan 74LS688. Dari gambar di atas terlihat bahwa input pada 74HC574 yang berlabel D2 sampai D8 terhubung dengan B0 sampai B7 yang merupakan output dari rangkaian osilator. Sedangkan clock dari 74HC574 terhubung dengan clock dari 74LS14. Hal ini dimaksudkan agar pada setiap detak pulsa yang dihasilkan dari 74HC14 akan *drive* data dari label D (input) menuju label Q (output) pada 74HC574. Adapun output dari

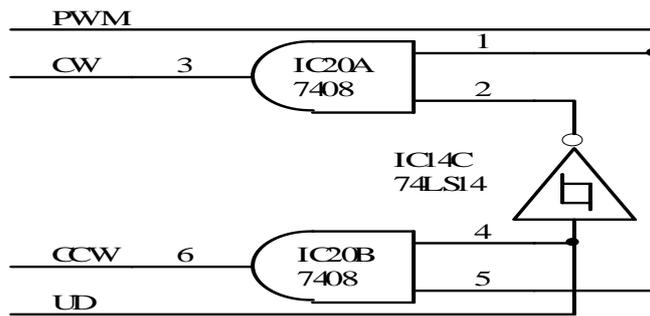
74HC574 terhubung dengan 74LS688 pada label P. Sedangkan pada label Q (C0 hingga C6) terhubung dengan PC. Apabila data dari 74HC574 (label P) sama kondisinya dengan data yang diberikan oleh PC (label Q) maka data yang sama tersebut akan dikeluarkan melalui pin 9 (dengan label SD) pada 74LS688 dengan kondisi *terbalik (inverter)*.

Selanjutnya, data dari pin 9 (label SD) pada 74LS688 yang berkondisi terbalik menuju 74LS74 (*D Flip Flop*) pada pin 1 dengan kondisi 'terbalik kembali' sehingga akan sesuai dengan data yang *diinputkan* dari PC.



Gambar 3.8. Rangkaian Reset dan Penghasil PWM

Dari gambar di atas terlihat bahwa kondisi CLK dan D pada 74LS74 selalu dalam kondisi high (terhubung dengan VCC). Hal ini dimaksudkan agar output yang dihasilkan (Q) selalu high pula. Adapun label SD (pin 4) pada 74LS74 terhubung dengan output 74LS30 (*Nand Gate*). Hal ini dimaksudkan agar ketika semua data yang berasal dari rangkaian osilator (B0 hingga B6) berkondisi *high* masuk ke *Nand Gate* akan dikeluarkan dalam bentuk *low*. Adapun fungsi dari CD adalah meng-*clear*-kan data dari input D (apabila CD dalam kondisi *low*). Sedangkan fungsi SD di sini adalah mengaktifkan kembali ketika data telah mencapai cacahan maksimum (8 bit). Adapun data PWM akan diteruskan melalui 7408 (*And Gate*) seperti pada gambar:

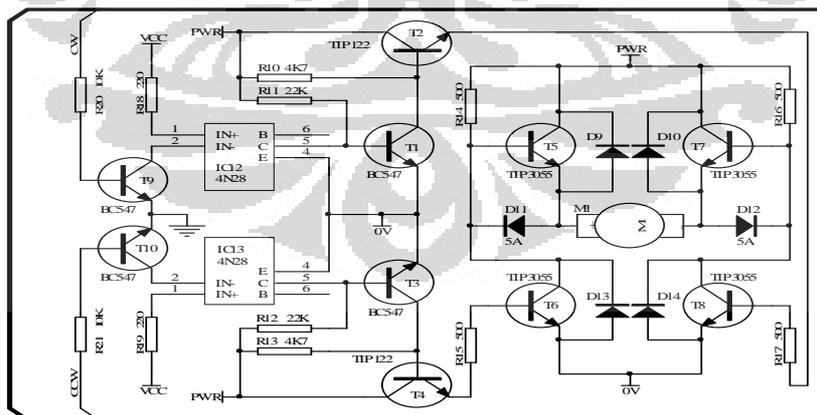


Gambar 3.9. Rangkaian CW/CCW

Dari gambar terlihat bahwa data PWM masuk ke salah satu input dari dua *Nand Gate*, sedangkan input yang lain berasal dari PC (label UD). Untuk dapat mengubah arah motor DC, diperlukan kondisi yang berbeda pada dua buah kutub yang dimiliki motor DC. Untuk itu maka diberikan kondisi yang berbeda pada salah satu input And Gate dengan menggunakan 74LS74 (inverter). Dengan demikian kondisi CW dan CCW akan selalu berbeda.

3.2.3. Rangkaian Penggerak Motor DC

Rangkaian penggerak motor yang digunakan menggunakan rangkaian transistor seperti pada gambar:



Gambar 3.10. Rangkaian Penggerak Motor DC

Seperti dibahas sebelumnya input motor tidak boleh berkondisi sama, atau dapat dikatakan bahwa kondisi CW dan CCW harus berbeda. Sebagai contoh misalkan CW

berkondisi 1 sedangkan CCW berkondisi 0 maka T9 akan aktif dan VCC akan mengalir menuju ground melalui IC12 4N28. Dengan mengalirnya tegangan pada 4N28, maka tegangan PWR (24 volt) mengalir melalui pin C menuju pin E (*saturasi*) pada 4N28. Dengan keadaan *saturasi* ini, maka mengakibatkan kondisi pada *basis* T1 akan *low* karena terhubung dengan pin C pada 4N28. Dengan kondisi *low* pada *basis* T1, maka transistor tersebut (T1) tidak aktif (*cutoff*) sehingga tegangan yang berasal dari PWR (24 volt) akan tertahan pada kaki *colector* T1. Dengan kondisi ini tegangan yang berasal dari PWR akan mengaktifkan *basis* pada T2. Aktifnya *basis* T2 mengakibatkan transistor tersebut akan aktif (*saturasi*) sehingga tegangan PWR mengalir menuju *emitor* pada T2. Selanjutnya tegangan dari T2 mengalir menuju kaki *basis* pada T8, sehingga transistor T8 menjadi aktif dan tegangan PWR langsung menuju *emitor* pada T8 dengan melewati T7 dikarenakan T7 dalam keadaan *off*. Di lain pihak, tegangan PWR juga menuju T5 dan mengaktifkan transistor tersebut sehingga tegangan yang melewati T5 mengalir menuju 0 volt melalui motor DC. Keadaan ini mengakibatkan kutub positif motor (+) menjadi lebih positif dibandingkan kutub negatifnya (-) sehingga motor berputar searah jarum jam.

Begitu pula sebaliknya jika CCW berkondisi 1 maka T10 akan aktif dan VCC akan mengalir menuju ground melalui IC13 4N28. Dengan mengalirnya tegangan pada 4N28, maka tegangan PWR (24 volt) mengalir melalui pin C menuju pin E (*saturasi*) pada 4N28. Dengan keadaan *saturasi* ini, maka mengakibatkan kondisi pada *basis* T3 akan *low* karena terhubung dengan pin C pada 4N28. Dengan kondisi *low* pada *basis* T3, maka transistor tersebut (T3) tidak aktif (*cutoff*) sehingga tegangan yang berasal dari PWR (24 volt) akan tertahan pada kaki *colector* T3. Dengan kondisi ini tegangan yang berasal dari PWR akan mengaktifkan *basis* pada T4. Aktifnya *basis* T4 mengakibatkan transistor tersebut akan aktif (*saturasi*) sehingga tegangan PWR mengalir menuju *emitor* pada T4. Selanjutnya tegangan dari T4 mengalir menuju kaki *basis* pada T6, sehingga transistor T6 menjadi aktif dan tegangan PWR langsung menuju *emitor* pada T6 dengan melewati T5 dikarenakan T5 dalam keadaan *off*. Di lain pihak, tegangan PWR juga menuju T7 dan mengaktifkan transistor tersebut sehingga tegangan yang melewati T7 mengalir menuju 0 volt melalui motor DC. Keadaan ini mengakibatkan kutub negatif motor (-) menjadi lebih positif dibandingkan kutub positifnya (+) sehingga motor berputar berlawanan arah jarum jam.

signal dengan kondisi yang berbeda. Adapun kedua output dari 74LS74 (Q dan \bar{Q}) diumpankan ke dua buah *Nand-Gate*. Hal ini dimaksudkan agar kedua output *Nand-Gate* selalu dalam kondisi berbeda. Perbedaan kondisi inilah yang akan digunakan untuk pembeda antara cacahan naik (*up counter*) dengan cacahan turun (*down counte*)

Adapun kedua output dari *Nand-Gate*,(TCU untuk *up counter* dan TCD untuk *down counter*) diumpankan ke IC 74LS193 yang merupakan *counter* 4 bit. Sedangkan counter yang dihasilkan sebesar 24 bit dengan menggunakan enam buah IC 74LS193.

Penggunaan 24 bit counter ini didasari dari jumlah cacahan yang dihasilkan motor serta karakteristik baut yang digunakan. Apabila:

$$1 \text{ putaran} = 100 \text{ mm} = 400 \text{ pulsa}$$

maka untuk menempuh keseluruhan panjang besi rel (100 cm) diperlukan:

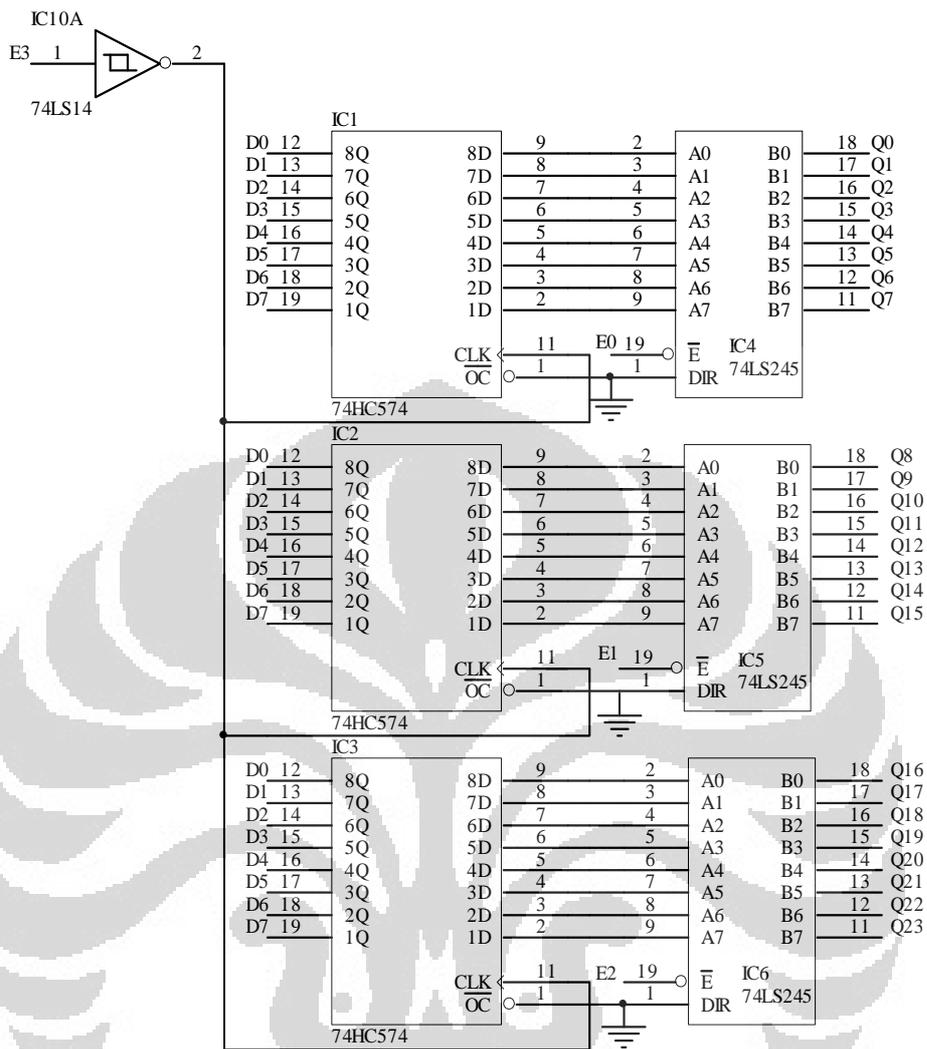
$$100 \text{ mm} \frac{1000 \text{ mm}}{100 \text{ mm}} = 1 \text{ putaran} \quad \frac{1000 \text{ mm}}{100 \text{ mm}} = 400 \text{ pulsa} \quad \frac{1000 \text{ mm}}{100 \text{ mm}}$$

$$1000\text{mm} = 10 \text{ putaran} = 4000 \text{ pulsa}$$

Adapun nilai maksimal untuk 24 bit adalah 16777216. Nilai ini amat sangat aman bila digunakan untuk menampung 4000 pulsa yang dihasilkan.

3.2.5. Rangkaian Latching

Rangkaian *latching* digunakan sebagai penyimpan data perhitungan yang dilakukan oleh counter, yang telah dibaca oleh sensor. Rangkaian *latching* ini digunakan karena, counter tersebut menghitung secara terus-menerus dan berubah-ubah dengan cepat mengikuti perputaran motor, oleh karena itu counter pun sulit untuk mengambil data dan data yang dihasilkan counter tidaklah begitu akurat Adapun rangkaian *latching* yang digunakan adalah sebagai berikut:

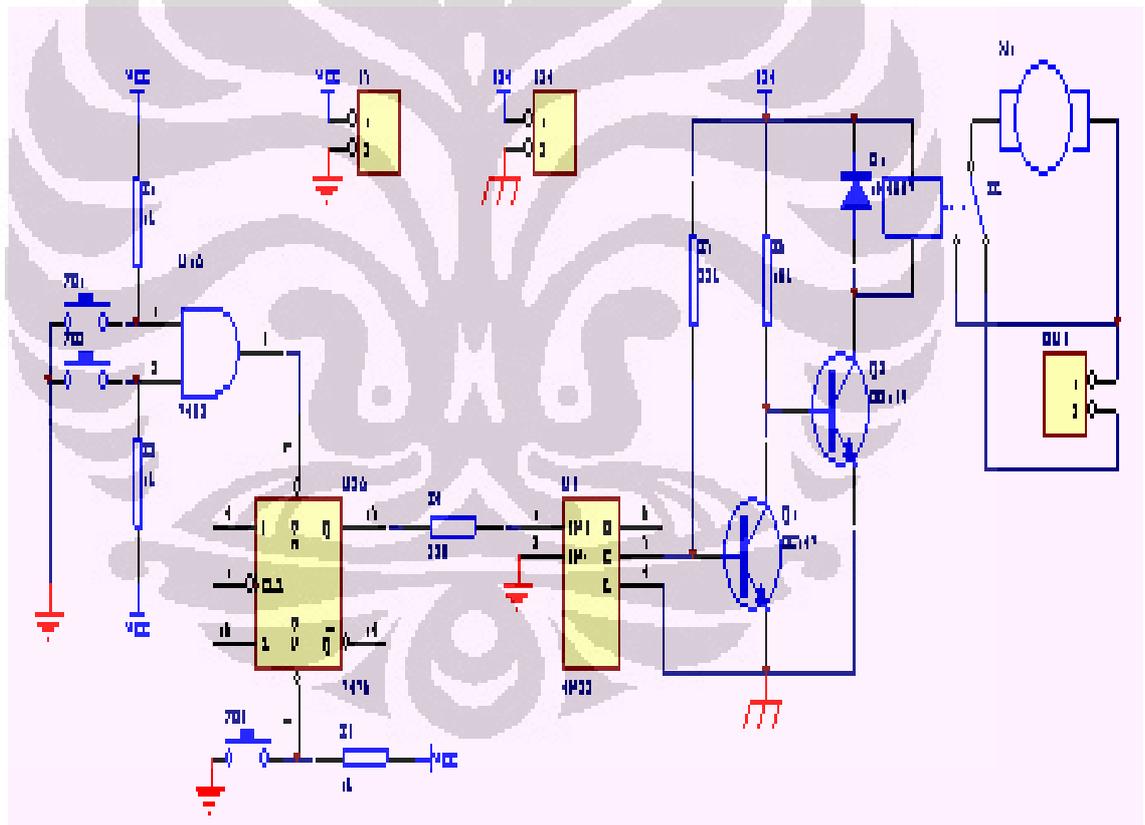


Gambar 3.12. Rangkaian *Latching*

Rangkaian *latching* ini menggunakan dua buah IC yaitu 74LS245 dan 74HC574. Jalur DIR pada 74LS245 terhubung ke ground, hal ini mengakibatkan aliran data berpindah dari lajur B menuju lajur A. Semua data counter 24 bit masuk jalur B pada 74LS245. Adapun jalur A terhubung dengan input 74HC574 (lajur D), sedangkan output 74HC574 menuju ke rangkaian *interfacing*. Data 24 bit yang masuk 74LS245 akan dialirkan menuju 74HC574 ketika *output enable* (E0 hingga E2) aktif (aktif low). Data tersebut diubah menjadi 8 bit dan dikirim secara bertahap menuju rangkaian *interfacing*. Sedangkan E3 berfungsi sebagai penentu kapan data dari 74LS245 dialirkan menuju rangkaian *interfacing* melalui 74HC574.

3.2.6 Rangkaian Switch

Rangkaian switch ini berfungsi untuk melakukan sistem pengereman, jika terjadi kelebihan dari pengendalian yang dilakukan. Bisa disebut juga rangkaian switch ini sebagai rangkaian pengaman. Dibuatnya rangkaian ini dikarenakan, motor yang dipakai mempunyai kecepatan yang tinggi yaitu sebesar 4000 rpm, oleh karena itu pengendalian yang dilakukan juga sangat cepat. Agar motor tidak mengalami kerusakan maka dibuatlah rangkaian ini, yang dipakai untuk menghentikan perputaran motor. Adapun rangkaian yang switch yang digunakan adalah sebagai berikut:



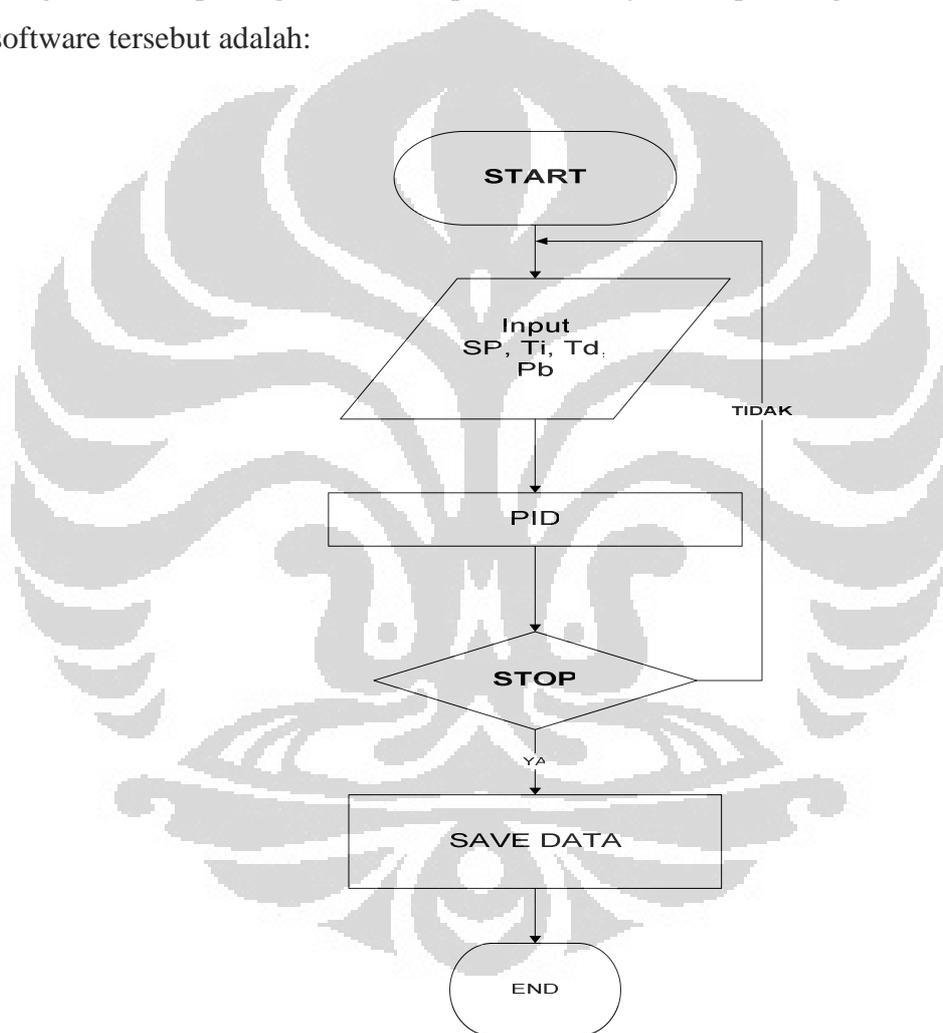
Gambar 3.13. Rangkaian Switch Motor DC

Pada saat switch 1 atau switch 2 ditekan, maka arus akan mengalir menuju IC 7483 yang berlogika AND, IC tersebut digunakan karena mempunyai gerbang logika yaitu

AND. Kemudian data masuk kedalam transistor dan dapat ditrigger, lalu kondisi relay dapat menyala, dan saat itu juga motor terhenti perputarannya.

3.3 Perancangan Software

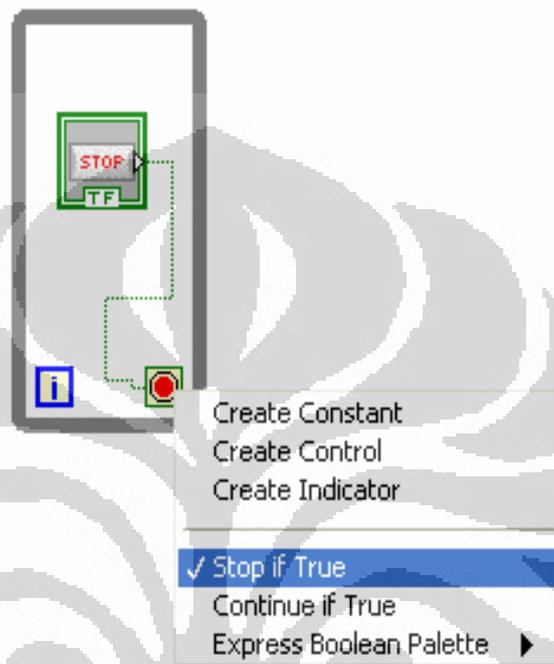
Pada sub bab ini penulis memasukan flow chart dari program, dan mengelompokkan menjadi beberapa bagian beserta pembahasannya. Adapun bagian dari perancangan software tersebut adalah:



Gambar 3.14. Flow Chart Program

3.3.1. Operasi Start/Stop

Operasi ini digunakan untuk memulai dan mengakhiri semua operasi yang dijalankan. Adapun instruksi yang digunakan adalah:



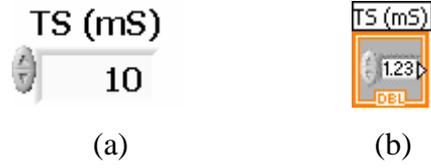
Gambar 3.15. Operasi Start/Stop

Pengoperasian start/stop menggunakan instruksi *while loop* dengan kondisi “stop if true”. Hal ini berarti bahwa program akan terus menerus melakukan *looping* sebelum *button* dalam kondisi *true* (ditekan), atau dengan kata lain program akan men-stop *looping* ketika *button* ditekan.

3.3.2. Operasi PID

Dalam pengaplikasian pada software, penulis menggunakan software LabVIEW 7.1, karena software ini lebih mudah digunakan dalam mengaplikasikan proses tersebut. Untuk tampilan pada PC, disediakan fasilitas untuk memasukkan komponen dalam PID *control*. Adapun komponen-komponen tersebut adalah TS (ms), SP (%), PB (%), Ti (ms), Td (ms). Komponen-komponen yang telah disebutkan berbentuk bilangan.

Sedangkan fasilitas pada LabVIEW 7.1 dalam memasukkan suatu bentuk bilangan digunakan *Numeric Control* pada bagian *Front Panel*.



Gambar 3.16.(a) Numeric Control pada Front Panel (b) Numeric Control pada Block Diagram

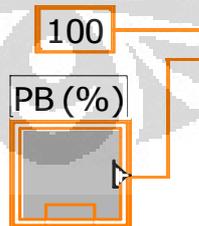
Dalam pengaplikasian software, nilai PB berbentuk persen (%). Bila melihat pada persamaan 3.2 maupun 3.3, nilai PB berasal dari nilai KP. Adapun persamaan awalnya adalah

$$PB \times KP = 100\% \dots\dots\dots(3.2)$$

sehingga

$$KP = \frac{100\%}{PB} \dots\dots\dots(3.3)$$

Adapun instruksi dalam LabVIEW adalah:



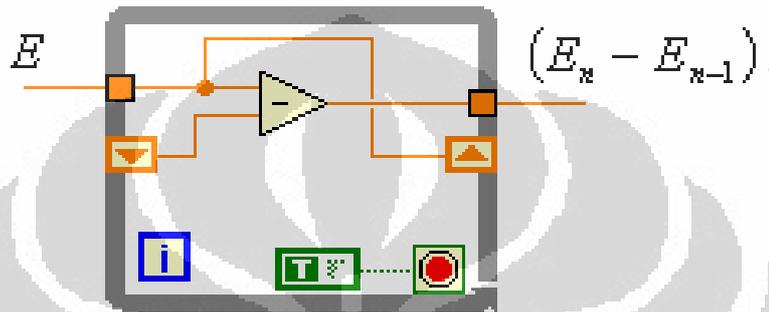
Gambar 3.17. Operasi dari Persamaan (3.3.)

Pada persamaan 3.2, unsur E mengalami perubahan dari persamaan 3.3. Pada bagian *integrator* berubah dari $\int E dt$ menjadi $\sum E$, dan pada bagian *differesiator*

berubah dari $\frac{dE}{dt}$ menjadi $(E_n - E_{n-1})$. Sedangkan E sendiri adalah *error* yang berasal dari persamaan:

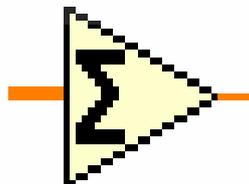
$$E = SP - PV \dots\dots\dots(3.4)$$

Program dari nilai $(E_n - E_{n-1})$ dalam LabVIEW adalah:



Gambar 3.18. Operasi dari Persamaan $(E_n - E_{n-1})$

Mula-mula nilai E masuk ke operasi *while loop* selanjutnya nilai “E awal” disimpan dalam *shift register*, adapun nilai E sebelum disimpan dalam *shift register*, dikurangi dengan data lama yang terdapat pada *shift register*. Untuk *looping* yang pertama, data pada *shift register* adalah 0 karena belum terisi oleh nilai E. Sedangkan untuk *looping* kedua dan seterusnya data pada *shift register* akan terisi oleh nilai E_{n-1} yaitu nilai E sebelum “E terbaru” (E_n). Operasi dari \sum dalam LabVIEW menggunakan *add array element*:

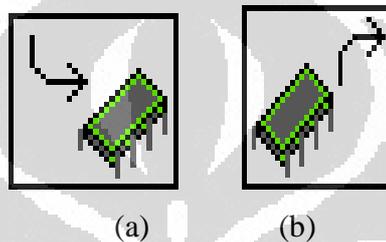


Gambar 3.19. Operasi *add array element*

Adapun fungsi dari operasi tersebut adalah menambahkan semua element array pada instruksi sebelumnya.

3.3.3. Program Komunikasi Paralel

Untuk mengakses suatu komunikasi paralel digunakan instruksi *out port* dan *in port*. Instruksi *out port* digunakan untuk mengirim data, sedangkan instruksi *in port* digunakan untuk mengambil data.



Gambar 3.20. (a) Outport (b) Inport

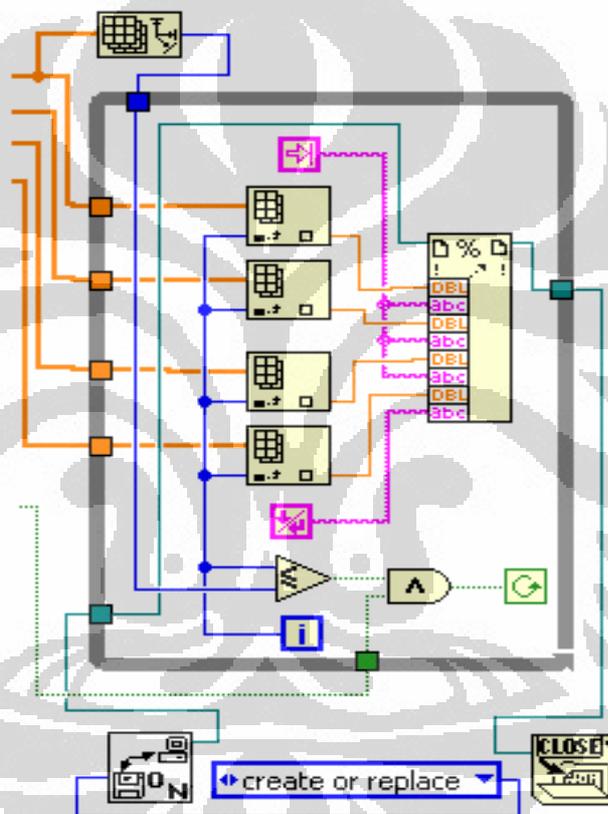
Adapun data-data yang dikirim memiliki alamat masing-masing sesuai dengan pin pada port printer sesuai dengan tabel 3.2. Alamat pada port printer terhubung dengan hardware dengan fungsi yang telah ditentukan.

Tabel 3.3. Koneksi Port Pararel dengan Hardware

Alamat	Koneksi	Fungsi
888	D0 – D7	Pengiriman Data Power
889	S3 – S6	Pembacaan Data Counter
890	C0 – C3	Pengaturan Enable

3.3.4. Program Penyimpanan Data

Yang dimaksud penyimpanan data disini adalah penyimpanan data hasil dari pengolahan program PID ke dalam file excel (.xls). Adapun data-data yang akan disimpan dalam file excel adalah data-data *Set Point (SP)*, *Proses Variable (PV)*, *Manipulated Variable (MV)* serta nilai *Time* (dalam second) yang didapat dari hasil kali antara iterasi dengan *Time Sampling (TS)*.



Gambar 3.21. Program Penyimpanan Data

Adapun data-data hasil pengolahan PID dimasukkan ke dalam dimensi *array* sedangkan *index array* terhubung ke iterasi. Sedangkan *element array* yang dihasilkan dimasukkan ke dalam format file excel (.xls).

BAB 4

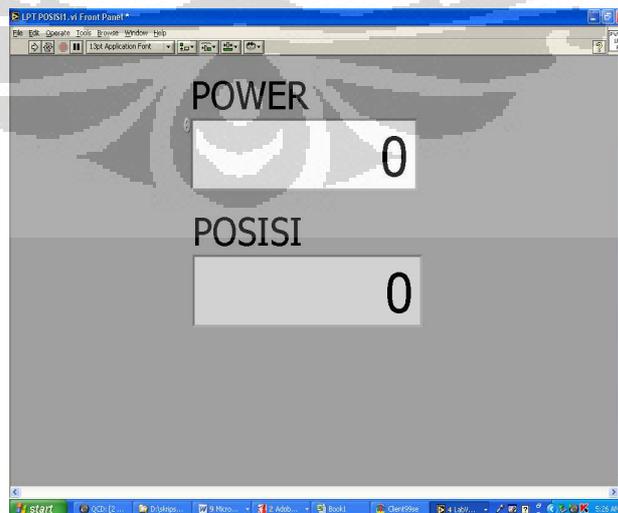
HASIL PERCOBAAN DAN PERHITUNGAN

Setelah dilakukan pengerjaan keseluruhan sistem, maka perlu dilakukan pengujian alat serta penganalisaan terhadap alat yang telah di buat, apakah sistem sudah bekerja dengan baik atau tidak, dan dari percobaan tersebut kita dapat mengetahui berbagai macam tingkat ketelitian yang mampu dihasilkan oleh hardware terhadap porogram yang telah dirancang. Pengujian-pengujian tersebut meliputi :

- Pengujian Pulsa Sensor
- Pengujian Perpindahan

4.2. Pengujian Pulsa Sensor

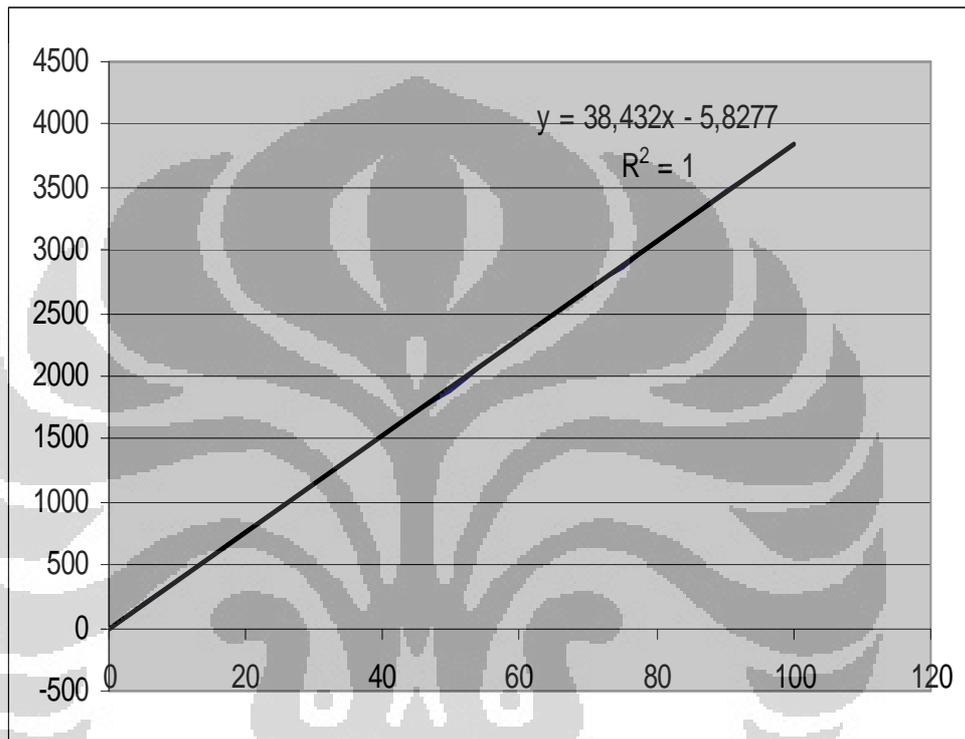
Pengujian pulsa disini bertujuan untuk mengetahui banyaknya pulsa yang dihasilkan setiap pergeseran 5 cm pada motor DC. Sedangkan metode yang dilakukan dengan menggunakan sebuah form uji coba dimana program tersebut hanyalam untuk menguji jumlah pulsa yang dihasilkan motor DC. Pengujian dilakukan dengan menarik besi Alloy setiap kenaikan 5 cm dan penurunan setiap 5 cm juga. Dari hasil percobaan diatas didapatkan data sebagai berikut:



Gambar 4.1. *Form* Pengujian Pulsa Sensor

Pengujian dilakukan dengan memutar lengan dari motor DC sebanyak 200 kali sehingga dan mencatat berapa pulsa yang dihasilkan dalam tiap putaran. Adapun jumlah pulsa yang dihasilkan dapat diamati melalui form pada kolom “posisi” melalui PC.

Dari pengujian didapatkan data yang dapat dilihat pada bagian lampiran. Adapun hasil pengolahan data didapat grafik sebagai berikut:



Grafik 4.2. Posisi Terhadap Pulsa

Dari data yang telah didapat dari pengujian perpindahan, dapat diketahui nilai rata-rata perpindahan yang dilakukan tiap putaran, dari persamaan:

$$\bar{y} = \frac{\sum y}{n} \dots\dots\dots(4.1)$$

Nilai n merupakan jumlah data yang didapat atau dalam percobaan ini adalah jumlah putaran yang dilakukan. Sehingga dapat diperoleh nilai rata-rata dari pulsa yang dihasilkan dari setiap pengukuran. Hasil persamaan ini merupakan hasil dari percobaan

pada pergeseran kotak besi sebesar 5 cm. Sehingga dalam 1 putaran katrol terjadi perpindahan sejauh 10 cm.

Hal ini juga dapat diartikan bahwa dalam 1 putaran terjadi rata-rata perpindahan sejauh 10 cm. Dari gambar grafik didapat persamaan sebagai berikut:

$$Y = 38,432 X + 5,8277 \dots \dots \dots \text{ (pers 4.2)}$$

Dari persamaan diatas kita dapat melihat bahwa gradien yang didapat bernilai 38,432. Gradien tersebut digunakan untuk mengkonfersi data dalam program, agar posisi yang dikendalikan dapat sesuai dengan pengukuran pada penggaris. Kemungkinan terjadi ketidak akuratan disebabkan karena adanya data yang tidak stabil pada saat pengambilan data sehingga grafik yang diperoleh tidak begitu linier.

4.2. Pengujian Perpindahan

Pengujian perpindahan disini bertujuan untuk mengetahui ketepatan perpindahan dari perputaran yang dilakukan motor DC untuk menggerakkan kotak besi. Adapun metode yang digunakan adalah dengan mengubah set point yang terdapat pada program, dan memakai pengendalian PID dengan memasukan nilai T_i , T_d , dan P_b yang telah didapat. Dari program tersebut dimasukan angka kedalam set point sebesar 90, dan 10. Pengujian ini dilakukan berulang-ulang, masing-masing sebanyak 10 kali, kemudian dilakukan lagi secara bertahap dengan menurunkan nilai maksimum pengendalian sebanyak 10 cm. Percobaan ini dilakukan untuk mendapatkan *error* yang dihasilkan, dengan demikian kita dapat menganalisa hasil yang telah didapat. Dari percobaan diatas telah didapat hasil grafik dan data sebagai berikut:

Table 4.2. Data Perpindahan

NO	POSISI	PENGUKURAN (cm)	BEDA (cm)	ERROR (%)
1	10	10.5	0.5	5.0
2	90	90	0	0.0
3	10	10.3	0.3	3.0
4	90	90.2	0.2	0.2
5	10	10.2	0.2	2.0
6	90	90.3	0.3	0.3
7	10	10.3	0.3	3.0
8	90	90.2	0.2	0.2
9	10	10	0	0.0
10	90	90.1	0.1	0.1
11	10	9.8	0.2	2.0
12	90	89.8	0.2	0.2
13	10	10.2	0.2	2.0
14	90	89.5	0.5	0.6
15	10	10.2	0.2	2.0
16	90	90.5	0.5	0.6
17	10	9.9	0.1	1.0
18	90	90.5	0.5	0.6
19	10	10.3	0.3	3.0
20	90	90.5	0.5	0.6

Dari data yang telah didapat dapat dilihat bahwa masih ada error yang dihasilkan. Hal ini diakibatkan karena adanya kesalahan pada konstruksi mekanik yang mengakibatkan terjadinya pergeseran dari set point yang diinginkan. Tetapi kesalahan yang didapat sangatlah kecil. Dari grafik diatas dapat diketahui bahwa persamaan yang didapat adalah:

Dari data yang telah didapat dari pengujian perpindahan, dapat diketahui nilai kesalahan (*error*) perpindahan yang dilakukan tiap pergeseran, didapatkan dengan cara mengurangi antara posisi dan pengukuran, karena kita hanya ingin mengetahui perbedaan pergeserannya maka nilai yang di hasilkan selalu absolute, dalam artian positif dan negatifnya tidak berpengaruh.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah menyelesaikan perancangan sistem serta pengujian terhadap sistem tersebut, maka penulis dapat mengambil suatu kesimpulan bahwa :

1. Motor yang digunakan adalah motor dengan kecepatan 4000 rpm DC.
2. Pada pembuatan mekanik, tidak menggunakan besi stenlis yang berulir, melainkan besi stenlis polos.
3. Pegas yang digunakan berfungsi meredam gaya tarik secara langsung, yang dapat mengakibatkan kerusakan pada drat.
4. Untuk mengatur kecepatan motor DC digunakan rangkaian PWM.
5. Untuk menghitung banyaknya pulsa yang dihasilkan oleh sensor pada motor dc, digunakan rangkaian pengkondisi sinyal, dan rangkaian *counter*.
6. Untuk menahan data yang telah dihitung oleh *counter*, maka digunakan rangkaian *latching*.
7. Sensor yang digunakan berupa *shaft encoder*, yang dapat menghasilkan pulsa ketika dibaca oleh rangkaian *optocoupler*.
8. Jumlah pulsa yang didapat pada pengendalian sepanjang 100 cm adalah 4000 pulsa.
9. Hasil rata-rata penjumlahan data dari percobaan secara manual, digunakan untuk mencari nilai konversi, nilai konfersi yang didapat 38,432.
10. Pada hasil percobaan didapat perbedaan pengukuran, dengan set point yang diinginkan (*error*) tidak begitu jauh berbeda.
11. Untuk mendapatkan nilai yang akurat dalam pengendalian, kita harus memasukan nilai PID yang sesuai agar tidak terjadi overshoot.

5.2. Saran

Pada kesempatan kali ini penulis dapat memberikan saran-saran untuk penelitian pembuatan sistem pengendali posisi berikutnya antara lain sebagai berikut:

1. Penyempurnaan terhadap pengukuran dari konstruksi mekanik, dengan cara melihat dengan cermat hasil pengukuran yang sudah didapat dari hasil pengontrolan.
2. Pembuatan konstruksi mekanik harus sepresisi mungkin untuk menghindari terjadinya gangguan dalam pergerakan perpindahan. Untuk mendapatkan hasil yang stabil (PV yang diinginkan tanpa Overshot) dengan T_s yang kecil dianjurkan untuk terlebih dahulu menggunakan metode literature dalam memasukkan nilai-nilai ke dalam kolom yang disediakan.

DAFTAR PUSTAKA

Tipler, Fisika Dasar II, Erlangga, Jakarta, 1994

1. Malvino, Prinsip-Prinsip Elektronika, Erlangga, Jakarta, 1995.
2. <http://www.manningsrfid.com/pdfs/IDSpecs.pdf>
3. datasheet AT89S52 , <http://www.alldatasheet.com/search/AT89S52.pdf>
4. Novarina Rika, Pengendali Posisi Berbasis PC, Laporan TA Instrument, 2004.
5. Tokheim, Roger L., "Digital Electronics", Prentice-Hall International, Inc.
6. Ikhsan, Pengendali Posisi Berbasis PC dengan Menggunakan LabView, Laporan Tugas Akhir Instrument, 2002.
7. [http:// www. Wikipedia.com](http://www.Wikipedia.com). 15 Desember 2007.



Tabel L1.2: Data Error pada pengukuran antara 10 hingga 90

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR (%)
1	10	10,5	0,5	5,0
2	90	90	0	0,0
3	10	10,3	0,3	3,0
4	90	90,2	0,2	0,2
5	10	10,2	0,2	2,0
6	90	90,3	0,3	0,3
7	10	10,3	0,3	3,0
8	90	90,2	0,2	0,2
9	10	10	0	0,0
10	90	90,1	0,1	0,1
11	10	9,8	0,2	2,0
12	90	89,8	0,2	0,2
13	10	10,2	0,2	2,0
14	90	89,5	0,5	0,6
15	10	10,2	0,2	2,0
16	90	90,5	0,5	0,6
17	10	9,9	0,1	1,0
18	90	90,5	0,5	0,6
19	10	10,3	0,3	3,0
20	90	90,5	0,5	0,6

Tabel L1.3: Data Error pada pengukuran antara 10 hingga 80

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR (%)
1	10	10,3	0,3	3,0
2	80	80,5	0,5	0,6
3	10	9,7	0,3	3,0
4	80	80,2	0,2	0,3
5	10	10,4	0,4	4,0
6	80	80,8	0,8	1,0
7	10	10	0	0,0
8	80	80,2	0,2	0,3
9	10	9,9	0,1	1,0
10	80	79,7	0,3	0,4
11	10	9,9	0,1	1,0
12	80	80,4	0,4	0,5
13	10	9,8	0,2	2,0
14	80	79,8	0,2	0,3
15	10	10,1	0,1	1,0
16	80	80,5	0,5	0,6
17	10	9,8	0,2	2,0
18	80	79,7	0,3	0,4
19	10	9,9	0,1	1,0
20	80	80,1	0,1	0,1

Tabel L1.4: Data Error pada pengukuran antara 10 hingga 70

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR
1	10	10,3	0,3	3,0
2	70	69,9	0,1	0,1
3	10	10,2	0,2	2,0
4	70	70,2	0,2	0,3
5	10	10,2	0,2	2,0
6	70	70,3	0,3	0,4
7	10	10,2	0,2	2,0
8	70	70,4	0,4	0,6
9	10	10,3	0,3	3,0
10	70	70,2	0,2	0,3
11	10	9,9	0,1	1,0
12	70	70,1	0,1	0,1
13	10	9,9	0,1	1,0
14	70	70,4	0,4	0,6
15	10	9,8	0,2	2,0
16	70	69,9	0,1	0,1
17	10	10,2	0,2	2,0
18	70	70,2	0,2	0,3
19	10	10	0	0,0
20	70	70,5	0,5	0,7

Tabel L1.5: Data Error pada pengukuran antara 10 hingga 60

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR
1	10	10,3	0,3	3,0
2	60	60,5	0,5	0,8
3	10	9,9	0,1	1,0
4	60	59,8	0,2	0,3
5	10	10,2	0,2	2,0
6	60	60,3	0,3	0,5
7	10	10,2	0,2	2,0
8	60	60,4	0,4	0,7
9	10	9,9	0,1	1,0
10	60	60,5	0,5	0,8
11	10	10,2	0,2	2,0
12	60	60,3	0,3	0,5
13	10	9,9	0,1	1,0
14	60	60	0	0,0
15	10	10,2	0,2	2,0
16	60	60,2	0,2	0,3
17	10	10	0	0,0
18	60	60,4	0,4	0,7
19	10	10,2	0,2	2,0
20	60	60,3	0,3	0,5

Tabel L1.6: Data Error pada pengukuran antara 10 hingga 50

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR
1	10	10,2	0,2	2,0
2	50	48,9	1,1	2,2
3	10	10	0	0,0
4	50	49,2	0,8	1,6
5	10	9,9	0,1	1,0
6	50	50,1	0,1	0,2
7	10	9,9	0,1	1,0
8	50	50,3	0,3	0,6
9	10	10,2	0,2	2,0
10	50	49,8	0,2	0,4
11	10	10,2	0,2	2,0
12	50	50,1	0,1	0,2
13	10	9,9	0,1	1,0
14	50	48,7	1,3	2,6
15	10	9,9	0,1	1,0
16	50	50,2	0,2	0,4
17	10	10,2	0,2	2,0
18	50	50,2	0,2	0,4
19	10	10,1	0,1	1,0
20	50	49,8	0,2	0,4

Tabel L1.7: Data Error pada pengukuran antara 10 hingga 40

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR
1	10	9,97	0,03	0,3
2	40	40,3	0,3	0,7
3	10	9,9	0,1	1,0
4	40	40,2	0,2	0,5
5	10	9,9	0,1	1,0
6	40	40,2	0,2	0,5
7	10	9,9	0,1	1,0
8	40	40,3	0,3	0,7
9	10	9,9	0,1	1,0
10	40	40,2	0,2	0,5
11	10	10,2	0,2	2,0
12	40	40,3	0,3	0,7
13	10	10,2	0,2	2,0
14	40	40	0	0,0
15	10	10,2	0,2	2,0
16	40	40,3	0,3	0,7
17	10	10,2	0,2	2,0
18	40	40,3	0,3	0,7
19	10	10,2	0,2	2,0
20	40	40,3	0,3	0,7

Tabel L1.8: Data Error pada pengukuran antara 10 hingga 30

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR
1	10	10,2	0,2	2,0
2	30	31,2	1,2	4,0
3	10	9,9	0,1	1,0
4	30	31,1	1,1	3,7
5	10	9,9	0,1	1,0
6	30	30,5	0,5	1,7
7	10	10,2	0,2	2,0
8	30	30,3	0,3	1,0
9	10	10,2	0,2	2,0
10	30	30,3	0,3	1,0
11	10	9,9	0,1	1,0
12	30	30,5	0,5	1,7
13	10	9,9	0,1	1,0
14	30	30,3	0,3	1,0
15	10	9,9	0,1	1,0
16	30	30,3	0,3	1,0
17	10	9,9	0,1	1,0
18	30	30,3	0,3	1,0
19	10	10,2	0,2	2,0
20	30	30,4	0,4	1,3

Tabel L1.9: Data Error pada pengukuran antara 10 hingga 20

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR
1	10	9,9	0,1	1,0
2	20	21,2	1,2	6,0
3	10	9,9	0,1	1,0
4	20	21,1	1,1	5,5
5	10	10	0	0,0
6	20	21,2	1,2	6,0
7	10	10,2	0,2	2,0
8	20	20,9	0,9	4,5
9	10	10,3	0,3	3,0
10	20	20,9	0,9	4,5
11	10	10,2	0,2	2,0
12	20	20,9	0,9	4,5
13	10	10,3	0,3	3,0
14	20	21,1	1,1	5,5
15	10	10,2	0,2	2,0
16	20	20,9	0,9	4,5
17	10	10,3	0,3	3,0
18	20	20,8	0,8	4,0
19	10	9,9	0,1	1,0
20	20	20,7	0,7	3,5

Tabel L1.9: Data Error pada pengukuran secara random/ acak

NO	POSISI (cm)	PENGUKURAN (cm)	BEDA (cm)	ERROR
1	10	9,9	0,1	1,0
2	70	70,3	0,3	0,4
3	50	51,1	1,1	2,2
4	40	40,3	0,3	0,7
5	90	90,3	0,3	0,3
6	20	21,2	1,2	6,0
7	60	60,7	0,7	1,2
8	30	30,4	0,4	1,3
9	80	80	0	0,0
10	100	98,2	1,8	1,8

LAMPIRAN 2

PROGRAM PID PADA PENGENDALI POSISI

