

**PENGENDALI POSISI BERBASIS PC DENGAN
METODE PID**



Oleh :
YUNIARTI
2304210464

PROGRAM D3 INSTRUMENTASI INDUSTRI DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA
DEPOK
2007

PENGENDALI POSISI BERBASIS PC DENGAN METODE PID

**Laporan tugas akhir ini diajukan sebagai salah satu syarat untuk
memperoleh ijazah Diploma 3 Instrumentasi Industri Dan Elektronika**



**Oleh :
YUNIARTI
2304210464**

**PROGRAM D3 INSTRUMENTASI INDUSTRI DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA
DEPOK
2007**

LEMBAR PENGESAHAN

Nama : Yuniarti
NPM : 230421464
Jurusan : Instrumentasi Elektronika
Peminatan : Fisika Instrumentasi
Tanggal Sidang : 10 Juli 2007
Judul Tugas Akhir :

PENGENDALI POSISI BERBASIS PC DENGAN METODE PID

Tugas akhir ini telah diperiksa dan disetujui oleh :

PEMBIMBING I

(Arief Sudarmaji , M.T)

PENGUJI 1

PENGUJI 2

(Dr. Prawito)

(Supriyanto , Ssi)

KATA PENGANTAR

Dengan mengucapkan syukur kepada Allah SWT yang dengan izin dan limpahan rahmat juga karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini. Penyusunan tugas akhir ini merupakan syarat yang harus dipenuhi untuk menyelesaikan studi pada program Diploma 3 Instrumentasi Elektronika, Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia. Dengan judul "Pengendali Posisi berbasis PC dengan metode PID"

Dalam pembuatan tugas akhir ini, penulis memperoleh banyak bantuan serta dukungan dari berbagai pihak sehingga dapat diselesaikan dengan baik. Pada kesempatan ini penulis ingin menyampaikan rasa terimakasih kepada :

1. Allah SWT atas izin dan limpahan Rahmat serta karunia-Nya yang tak terkira, dan hanya dengan izinnya semua dapat terselesaikan.
2. Kepada Orang tua yang tak pernah lelah telah memberikan do'a dan motivasi kepada penulis sehingga terselesaikannya tugas akhir ini.
3. Bapak Dr. Prawito , selaku Ketua Program DIII Instrumentasi Elektronika.
4. Bapak Arief Sudarmaji, MT selaku pembimbing tugas akhir. Terima kasih atas arahan dan waktu yang telah diberikan.
5. Bapak Surya Darma, Msi selaku Koordinator Tugas Akhir.
6. Temen2 seperjuangan Rika, Irma, Hagi, Yayan, Dewi, Tya, Sang2, Seno, Agung, Yudith, Cahyo, Aryo, Abdul, Ludy, Fajar, Usman, Lindra, Wulan, Siska, Anak2Sailor(nana,desti,dini,risca,illa,maya), anak2 kontrakan

belakang (gege, wahyu, rizal, ichan, cakra, weduz, syamsul, beny),Anak2 kukel (haeril, hamdan, vai, slamet, rachmat, joker, cungk's, alet, ridho, mbul), Qthing'01, Arip'03, Mas Tedja (Thanx buat bantuannya) dan temen-temen yang belum disebut maaf ya.

7. Kawan – kawan di DELIMA. Cthee, Era, Upeh, Ambon, Qthing. Yang telah memberikan dorongan dan semangat biar kita lulus bareng.
8. Orang - orang yang telah mengisi hati Qu selama di Instrumen (Ivan'02, Edo'03, Sofyan'99, Seno'99). Thanx ya dah beri warna buat hidup Qu.
9. Temen – temen di Metro TV (Pa Barit, Udin, Ellon, Yudi, Riza, Ferry, SudirBoy, Mas Tommy) ma kasih buat nasehat dan pelajarannya itu sangat berharga.
10. Pihak-pihak lain yang tidak dapat penulis sebut satu persatu.

Penulis menyadari masih terdapat banyak kekurangan dalam penulisan tugas akhir ini, maka penulis berharap adanya masukan yang membangun berupa pendapat, saran dan kritik sehingga penulis dapat memperbaikinya

Semoga Laporan Tugas Akhir ini dapat bermanfaat bagi semua pihak yang membacanya, serta penulis sebagai penyusun khususnya, terlebih lagi bagi perkembangan ilmu pengetahuan.

Jakarta, Juli 2007

Penulis

DAFTAR ISI

Lembar Pengesahan.....	i
Kata Pengantar.....	ii
Daftar Isi.....	iv
Daftar Gambar.....	vi
Daftar Tabel.....	viii
Bab 1. Pendahuluan.....	1
1.1. Tujuan.....	2
1.2. Batasan Masalah.....	2
1.3. Deskripsi Singkat.....	3
1.4. Metode Penulisan.....	4
Bab 2. Teori Dasar.....	6
2.1. Motor DC.....	6
2.2. PWM.....	9
2.3. Drat.....	11
2.4. Sensor.....	11
2.4.1. Optocoupler.....	12
2.4.2. Shaftencoder.....	13
2.5. Prinsip Dasar PID.....	14

2.5.1. Open Loop Control Sistem.....	15
2.5.2. Closed Loop Control Sistem.....	16
2.5.2.1. Pengendali Propotional.....	17
2.5.2.2. Pengendali Integral.....	18
2.5.2.3. Pengendali Differensial.....	19
2.5.2.4. Pengendali PID.....	21
2.5.2.5. Pengendali PID struktur Paralel.....	22
2.5.2.6. Pengendali PID struktur Mix.....	23
2.5.2.7. Pengendali PID Struktur Seri.....	24
Bab 3. Analisa Rangkaian Dan Program.....	25
3.1. Rangkaian Interface.....	25
3.2. Konfigurasi Port Paralel.....	27
3.3. Analisa Program.....	30
3.3.1. Kontrol Secara Manual.....	30
3.3.2. Kontrol secara Auto.....	31
Bab 4. Hasil penelitian dan Analisa.....	33
4.1. Cara Pengambilan Data.....	33
4.2. Hasil Penelitian Dan Analisa.....	34
Bab 5. Kesimpulan Dan Saran.....	43
5.1. Kesimpulan.....	43
5.2. Saran.....	43

Daftar Pustaka

Lampiran

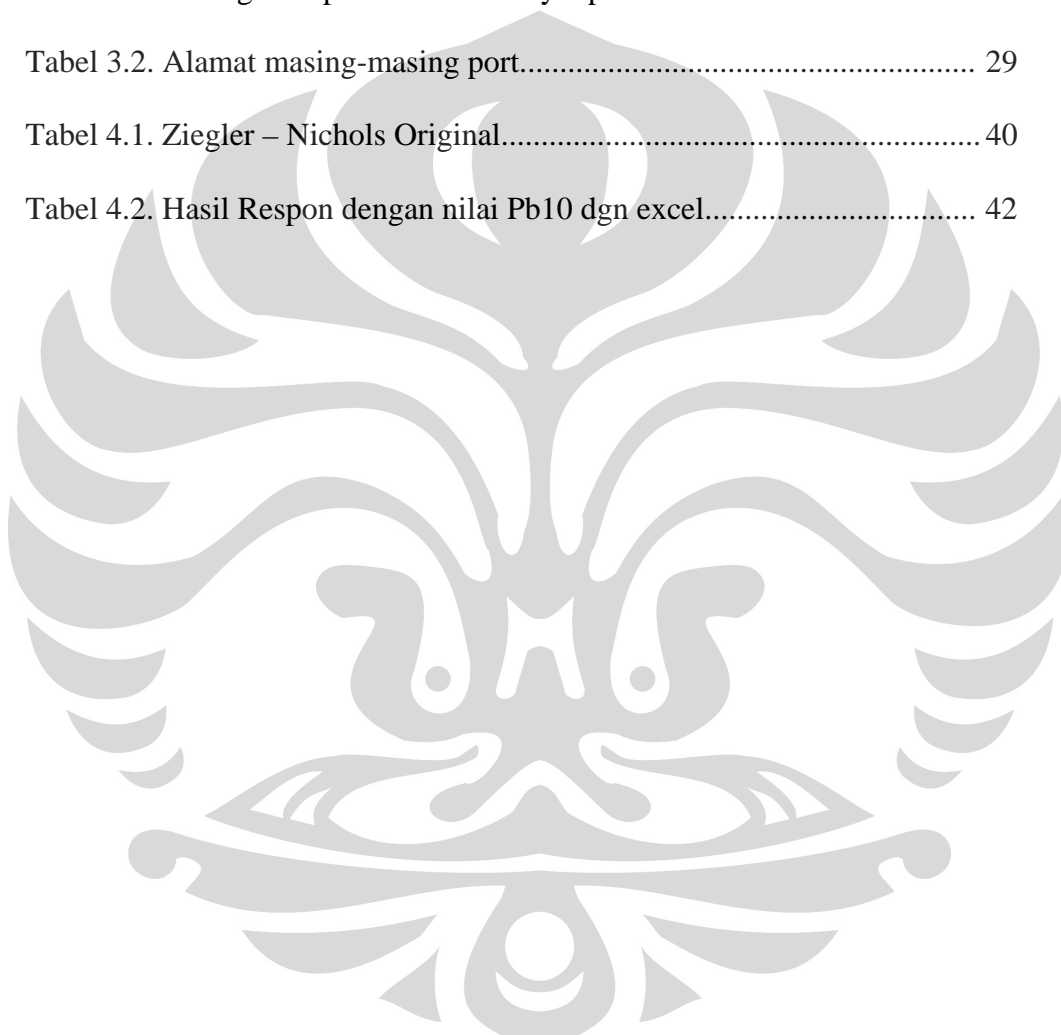
DAFTAR GAMBAR

Gambar 1.1. Blok diagram modul pengendali posisi	3
Gambar 1.2. Skematik modul pengendali posisi.....	4
Gambar 2.1. Gaya Medan Magnet.....	7
Gambar 2.2. Prinsip Motor DC.....	8
Gambar 2.3. Motor DC.....	9
Gambar 2.4. Arah Putaran Motor DC.....	9
Gambar 2.5. Cara Pengendalian Motor.....	10
Gambar 2.6. Ulir yang ada pada suatu Drat.....	11
Gambar 2.7. Gambar Skematik Dari OID.....	12
Gambar 2.8. Gambar Piringan Pada Shaft Encoder.....	13
Gambar 2.9. Gambar sensor Shaft Encoder.....	14
Gambar 2.9. Gambar Sistem Pengendali terbuka.....	15
Gambar 2.10. Sistem Pengendali Tertutup.....	16
Gambar 2.11. Grafik Respon Pengendali Propotional.....	17
Gambar 2.12. Grafik Respon Pengendali Integral.....	18
Gambar 2.13. Grafik Respon Pengendali Differensial.....	19
Gambar 2.14. Blok Diagram kontroler PID analog.....	21
Gambar 2.15. Grafik hubungan antara sinyal keluaran dan masukan.....	22
Gambar 2.16. Grafik respon Pengendali PID struktur Paralel.....	23

Gambar 2.17. Grafik respon Pengendali PID struktur Mix.....	24
Gambar 2.18. Grafik Respon Pengendali PID strukutr Seri.....	24
Gambar 3.1. Skematik Rangkaian Interface.....	27
Gambar 3.2. Skematik port paralel.....	28
Gambar 3.3. Flowchart secara manual.....	31
Gambar 3.4. Flowchart secara Auto.....	32
Gambar 4.1. Form Data Pada Program.....	34
Gambar 4.2. Respon Yang dihasilkan Pb 10.....	35
Gambar 4.3. Respon yang dihasilkan Pb 10 dgn data Excel.....	36
Gambar 4.3. Respon yang dihasilkan Pb 5 dengan data excel.....	37
Gambar 4.4. Respon yang dihasilkan Pb 2 dengan data excel.....	38
Gambar 4.5. Respon yang dihasilkan Pb 14.....	39
Gambar 4.6. Respon yang dihasilkan Pb 14 dgn data excel.....	40

DAFTAR TABEL

Tabel 3.1. Konfigurasi pin dan nama sinyal pd DB 25.....	28
Tabel 3.2. Alamat masing-masing port.....	29
Tabel 4.1. Ziegler – Nichols Original.....	40
Tabel 4.2. Hasil Respon dengan nilai Pb10 dgn excel.....	42



BAB 1

PENDAHULUAN

Pesatnya perkembangan dunia akhir-akhir ini tidak lepas dari semakin beragamnya perangkat instrument yang digunakan dalam menunjang aktivitas produksi dunia industri. Hal ini juga turut mendorong timbulnya berbagai macam alat instrumentasi dengan fungsi yang relative sama, namun dengan kualitas dan kehandalan yang berbeda. Dalam bab ini penulis akan membahas tentang latar belakang, tujuan, batasan masalah dan deskripsi singkat dari modul pengendali posisi yang penulis buat.

Dengan kemajuan teknologi dan perkembangan ilmu pengetahuan yang semakin pesat, maka tuntutan akan kebutuhan peralatan dan perlengkapan yang lebih cepat, sederhana, akurat dan ekonomis semakin meningkat yang kemudian menghasilkan perkembangan baru dalam perencanaan dan pemakaian. Semakin luas pula kebutuhan yang menuntut kita untuk meningkatkan kualitas ilmu pengetahuan khususnya dalam bidang pengendalian. Sistem kendali merupakan salah satu alat instrumentasi yang sangat penting dalam dunia industri. Sistem ini merupakan sebuah sistem yang terdiri atas salah satu atau beberapa peralatan yang berfungsi untuk mengendalikan sistem lain yang berhubungan dengan sebuah proses.

Sistem pengendalian telah banyak aplikasinya di bidang industri ataupun di bidang elektronika. Seperti pada industri tekstil dan perancangan robot sistem pengendalian khususnya pengendalian posisi sangatlah dibutuhkan. Karena semakin banyak kebutuhan yang menggunakan sistem pengendali maka dibutuhkan suatu sistem pengendali yang stabil, akurat, dan mempunyai kesalahan yang relatif kecil. Bila suatu sistem tersebut tidak berjalan sesuai dengan yang diharapkan, maka sistem pengendali ini dapat mengendalikan proses tersebut sehingga sistem dapat berjalan kembali sesuai dengan yang diharapkan.

Dari penjelasan diatas, penulis akan membuat suatu sistem pengendalian posisi dalam skala laboratorium. Dimana sistem pengendali posisi dibuat dengan menggunakan sistem digital sehingga kita cukup mengaturnya lewat komputer. Dan diharapkan sistem ini akan memperkecil kesulitan yang mungkin terjadi pada saat pengendalian posisi dan bekerja secara efektif dan efisien.

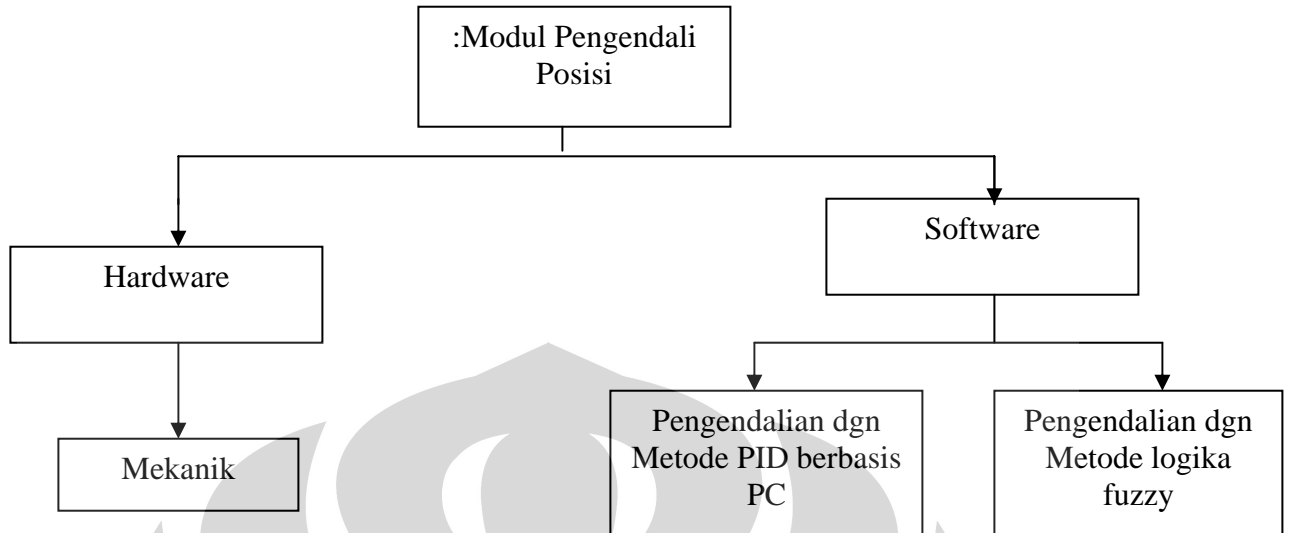
1.1. Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini antara lain adalah sebagai berikut :

- Untuk membuat suatu modul pengendali posisi yang berskala laboratorium dengan menggunakan metode PID berbasis PC.

1.2. Batasan Masalah

Pada modul ini ada dua komponen pendukung yaitu hardware dan software. Software pada modul ini digunakan untuk mengendalikan posisi dengan menggunakan komputer sebagai alat bantu. Ada 2 metode yang digunakan untuk mengendalikan modul ini yaitu Metode dengan menggunakan PID berbasis PC dan Metode dengan menggunakan logika fuzzy. Untuk itu penulis akan membahas lebih dalam tentang pengendalian posisi dengan metode PID berbasis PC. Untuk menghubungkan antara software dan hardware kita menggunakan rangkaian interface dan port paralel sebagai komunikasi. Untuk lebih jelas kita dapat lihat pada gambar dibawah ini :



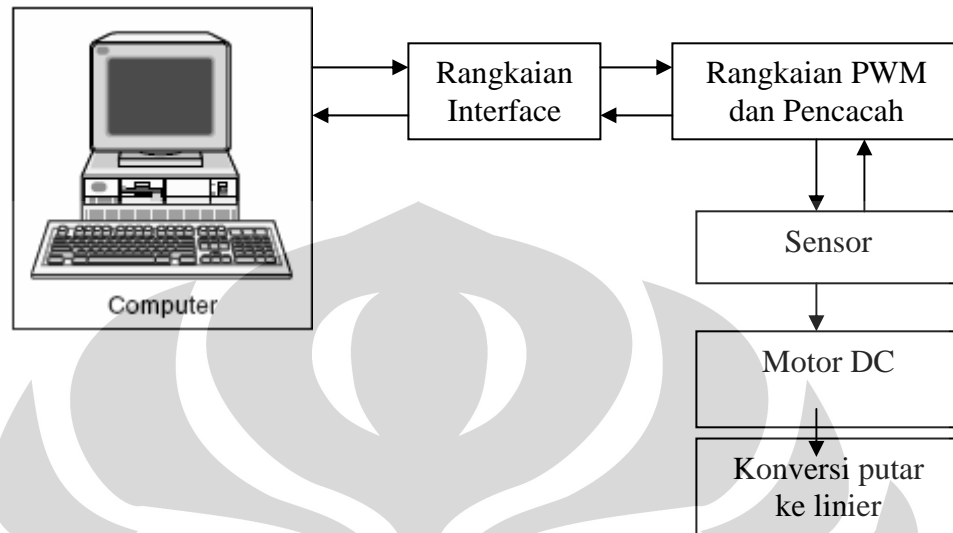
Gambar.1.1 Blok Diagram Modul Pengendalian Posisi

1.3. Deskripsi Singkat

Sistem ini bekerja jika komputer mengirim suatu sinyal PWM (Pulse Width Modulation) pada actuator, yang pada modul ini berupa motor DC, maka motor akan bergerak dan menggerakkan drat pada as sehingga konversi akan berputar ke linier. Saat drat berputar maka sensor akan membaca putaran dari drat dalam bentuk pulsa-pulsa listrik.

Data yang berupa pulsa-pulsa akan masuk ke rangkaian interface. Pada sistem ini rangkaian interface akan menerima data dari komputer dan mengirimnya ke rangkaian PWM sehingga kecepatan motor dapat diatur.

Sensor, motor DC, dan drat diletakkan pada satu as sehingga saat motor bergerak maka drat juga akan berputar. Saat drat berputar maka sensor akan menghitung pulsa yang dihasilkan oleh drat. Dari pulsa-pulsa ini kita akan melihat perpindahan posisi yang dihasilkan oleh motor.



Gambar.1.2 Skematik Modul Pengendalian Posisi

1.4. Metode Penulisan

Dalam penulisan penelitian ini, penulis membuat urutan cara penulisan yang secara garis besar diuraikan sebagai berikut :

1.4.1 Bab 1. Pendahuluan

Pada bab ini penulis akan membahas tentang latar belakang yang mendasari pembuatan dari modul ini. Lalu tujuan dari penelitian ini serta batasan masalah, deskripsi singkat tentang modul ini dan metode penulisan.

1.4.2 Bab 2. Teori Dasar

Pada bab ini akan membahas tentang teori dasar yang sangat diperlukan agar pembaca lebih mengerti tentang modul ini. Teori yang ditulis antara lain Motor DC, PWM, Drat, Sensor dan metode PID.

1.4.3 Bab 3. Cara kerja rangkaian dan listing program

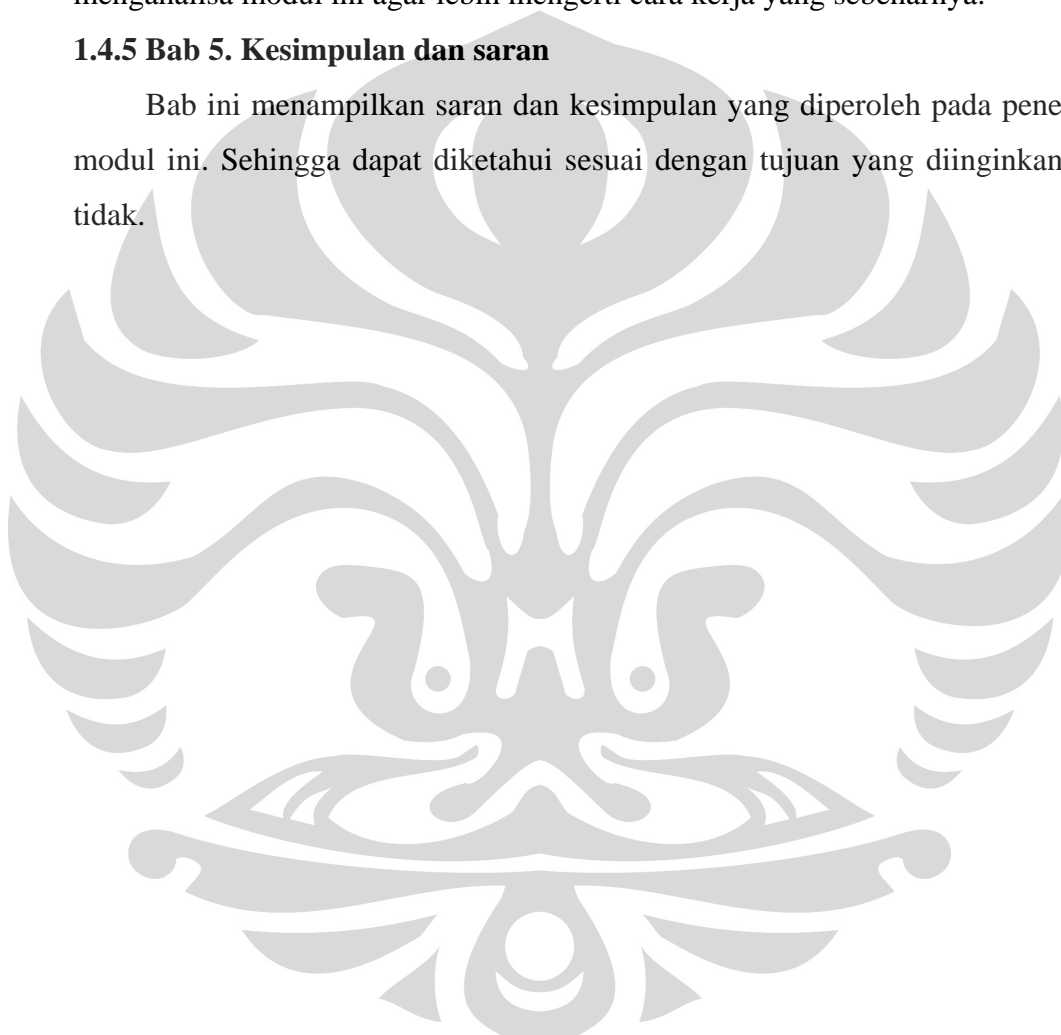
Bab ini akan menjelaskan cara kerja dari rangkaian interface serta listing program yang digunakan pada modul pengendali posisi ini.

1.4.4 Bab 4. Hasil dan Analisa

Bab ini akan membahas hasil yang diperoleh dari pengujian yang dilakukan pada modul ini yang dapat digunakan sebagai acuan dari modul ini. Serta menganalisa modul ini agar lebih mengerti cara kerja yang sebenarnya.

1.4.5 Bab 5. Kesimpulan dan saran

Bab ini menampilkan saran dan kesimpulan yang diperoleh pada penelitian modul ini. Sehingga dapat diketahui sesuai dengan tujuan yang diinginkan atau tidak.



BAB 2

TEORI DASAR

Perancangan modul pengendali posisi dengan menggunakan metode PID ini sebagaimana tercantum dalam tujuan penelitian. Ada beberapa pemahaman dasar yang sangat diperlukan agar kita lebih memahami modul ini. Beberapa pemahaman dasar tersebut antara lain : Motor DC, PWM (Pulse Width Modulation), Drat, Sensor dan Prinsip dasar PID.

2.1. Motor DC

Actuator adalah peralatan yang mengkonversi sinyal elektrik menjadi gerak mekanik. Bentuk umum dari actuator antara lain yaitu relay, solenoid dan motor. Pada modul ini kita menggunakan motor DC sebagai actuator, motor ini memiliki tingkat efisiensi yang tinggi dan mempunyai putaran kontinu yang lancar sehingga memudahkan untuk menghitung pulsanya.

Motor DC memiliki prinsip kerja yaitu suatu penghantar yang berarus listrik dan ditempatkan dalam suatu medan magnet maka penghantar tersebut akan mengalami gaya. Prinsip kerja motor membutuhkan :

1. Adanya garis-garis gaya medan magnet (fluks), antara kutub yang berada di stator.
2. Penghantar yang berarus listrik yang ditempatkan dalam medan magnet tersebut.
3. Pada penghantar akan timbul gaya.

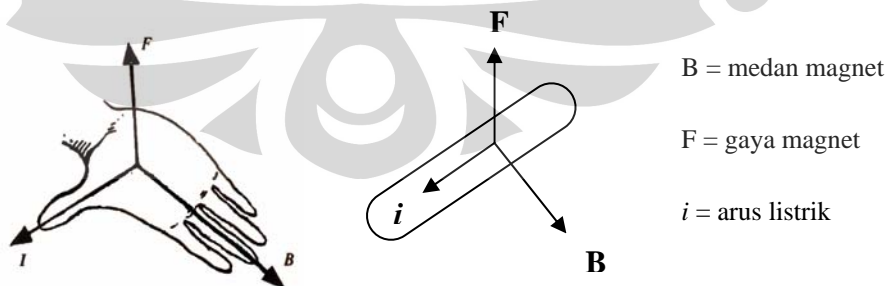
Pada Motor DC didesain untuk memanfaatkan gaya magnet untuk menghasilkan gerak berputar yang kontinyu dan disusun oleh komponen-komponen :

- Stator magnet digunakan sebagai penghasil gaya magnet permanen. Dibentuk menyesuaikan bentuk housing motor dengan setengah lingkaran atau satu lingkaran penuh.
- Armature coil digunakan sebagai kumpulan penghantar (konduktor) yang digulung sedemikian rupa hingga dapat menghasilkan torsi yang optimum. Duduk pada yoke yang dipasang permanen terhadap shaft.
- Commutator digunakan sebagai jalur masuk dan keluarnya arus listrik pada armature coil. Terbuat dari tembaga yang tersekat antar segmen oleh bahan isolator seperti mika.
- Brush digunakan sebagai medium penyalur arus listrik dari sumber listrik ke commutator. Terbuat dari tembaga atau carbon dan dedesain untuk lebih mudah aus dibandingkan dengan commutator.
- Bearing digunakan sebagai penyangga shaft pada housing motor.

Gaya yang dihasilkan motor dc tergantung pada :

- Kekuatan pada medan magnet.
- Besarnya arus yang mengalir pada penghantar.
- Panjang kawat penghantar yang berada dalam medan magnet.

Apabila panjang kumparan rotor L dialiri arus listrik sebesar I dan terletak diantara kutub magnet utara dan selatan dengan kerapatan fluks sebesar B , maka kumparan rotor tersebut mendapat gaya F sebesar :



Gambar 2.1. gaya medan magnet

$$F = B * I * L \quad (2-1)$$

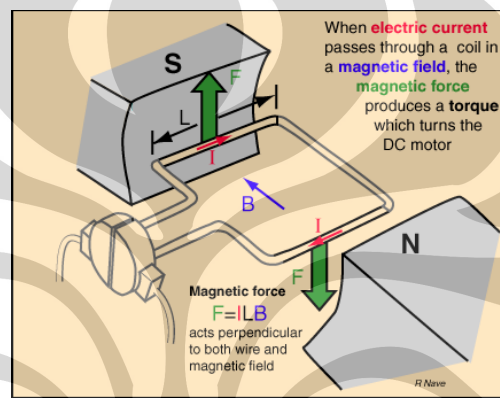
Keterangan :

F = Gaya Lorentz (Newton)

B = Kerapatan Fliks Magnet (Weber / m²)

I = Arus Listrik (Ampere)

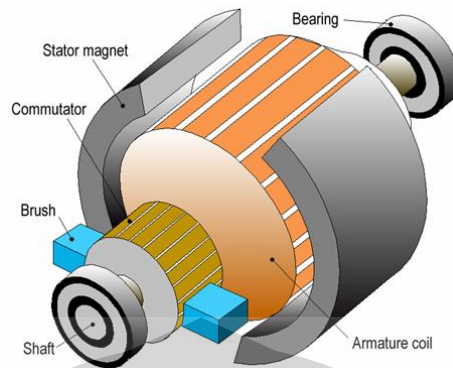
L = Panjang sisi kumparan rotor (m)



Gambar 2.2. Prinsip Motor DC

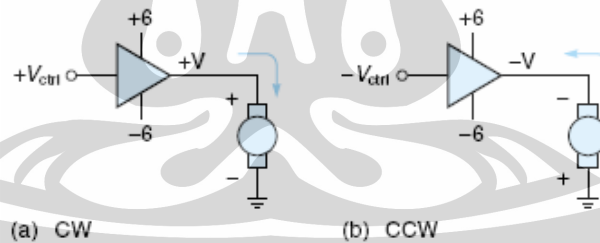
Adapun jenis-jenis motor DC berdasarkan cara memberikan arus eksitasi, motor arus searah dibagi dalam beberapa jenis :

- a. Motor arus searah dengan penguatan terpisah / bebas
- b. Motor arus searah dengan penguatan sendiri
 1. Motor arus searah shunt
 2. Motor arus searah seri
 3. Motor arus searah kompon pendek
 4. Motor arus searah kompon panjang



Gambar 2.3. motor DC

Untuk mengubah arah rotasi dari PM motor, polaritas dari tegangan yang digunakan adalah berlawanan. Satu cara untuk dapat melakukannya sehingga sebuah motor driver mampu mengeluarkan tegangan positif dan tegangan negatif. Ketika tegangannya positif dan ground maka motor akan bergerak searah jarum jam (CW). Ketika tegangannya negatif dan ground maka polaritas tegangan pada terminal motor berlawanan sehingga motor bergerak berlawanan arah jarum jam (CCW).



Gambar 2.4. Arah Putaran Motor DC

2.2. PWM (*Pulse Width Modulation*)

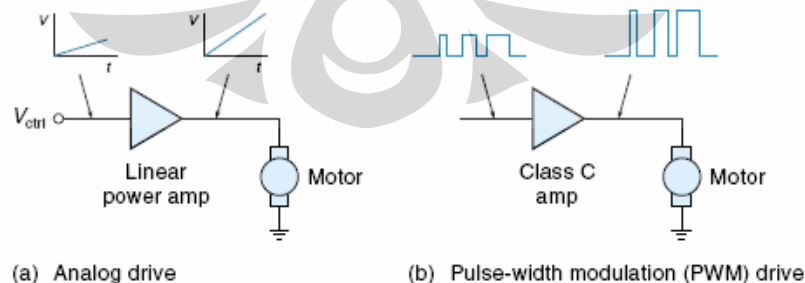
Pada dasarnya mesin listrik dapat berlaku sebagai motor maupun sebagai generator. Perbedaannya hanya terletak dalam konvensi dayanya. Generator adalah suatu mesin listrik yang mengubah daya masuk mekanik menjadi daya keluar listrik,

sedangkan motor mengubah daya masuk listrik menjadi daya keluar mekanik. Maka dengan membalik generator arus searah dimana tegangan menjadi sumber dan tegangan jangkar merupakan ggl lawan, mesin arus searah ini akan berlaku sebagai motor.

Terdapat beberapa cara untuk mengontrol kecepatan dari motor dc yaitu dengan mengubah jumlah kutub motor semakin banyak jumlah kutub motor maka kecepatan motor akan semakin tinggi, mengatur tahanan luar motor semakin kecil tahananannya maka kecepatan motor semakin tinggi dan mengubah tegangan yang diberikan pada motor semakin besar level tegangan yang diberikan maka kecepatan motor akan semakin tinggi.. Dan semua cara diatas disebut analog drive. Pada metode ini, sebuah penguatan power amplifier adalah linier dari controller dan tegangan analog diberikan ke motor.

Teknik yang lain untuk mengontrol sebuah motor dc adalah pulse-width modulation (PWM). Dalam sistem ini, power di suplai ke motor dalam bentuk pulsa dc pada tegangan tertentu. Lebar pulsa bervariasi untuk mengontrol kecepatan motor. Pulsa yang lebih lebar, tegangan dc rata-rata yang lebih tinggi diperbolehkan untuk motor. Frekuensi dari pulsa yang cukup tinggi dapat menginduktansi rata-rata motor sehingga dapat menggerakkan motor secara baik. Sistem ini memiliki 2 keuntungan bila dibandingkan dengan analog drive yaitu :

1. Power amplifier dapat menjadi tipe kelas C yang efisien.
2. DAC tidak dibutuhkan karena amplifier baik On atau Off dapat dikendalikan secara langsung dengan sinyal digital.

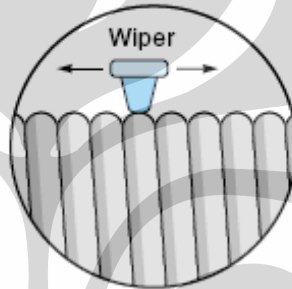


Gambar 2.5. Cara Pengendalian Motor

2.3. Dart

Drat adalah salah satu dari perangkat dari mekanik yang berfungsi sebagai konversi putar ke linier. Pada modul ini drat digunakan sebagai acuan untuk mengukur perpindahan jarak atau posisi yang terjadi saat motor DC digerakkan.

Menurut teori jarak dari ulir yang satu dengan yang lain sama dengan jarak perpindahan drat pada satu putaran. Untuk lebih jelasnya kita dapat melihat gambar 2-5 dibawah ini.



Gambar 2.6. Ulir yang ada pada suatu drat

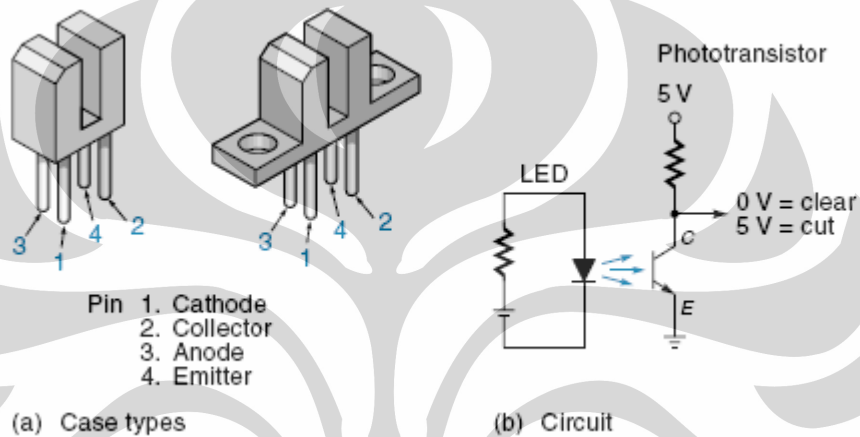
2.4. Sensor

Sensor adalah peralatan yang digunakan untuk mendeteksi ataupun mengukur ukuran dari sesuatu. Sensor umumnya dikategorikan menurut apa yang diukur dan sangat berperan penting dalam proses pengendalian manufaktur modern.

Sensor elektronik adalah sebuah komponen elektronik yang mengubah besaran bukan listrik menjadi besaran listrik (Brink and Flink, 1985). Beberapa jenis sensor ini diantaranya NTC, LM335 dan LDR. *Negative temperatur coefficient* (NTC) adalah sebuah resistor yang resistansinya dipengaruhi oleh suhu. Makin tinggi suhu, resistansinya makin bertambah kecil. Hubungan antara suhu dan resistansi tergantung pada komponen bersangkutan. Untuk itu setiap NTC yang digunakan sebagai alat ukur harus dikalibrasi terlebih dahulu (Cooper, 1985). Pada modul ini penulis menggunakan dua sensor yaitu sensor optocoupler dan sensor shaft encoder.

2.4.1. Sensor OID

Sensor ini disebut juga optoisolator atau isolator yang terganggu optik. OID menggabungkan LED dan fototransistor dalam satu kemasan. Gambar 2.7 menunjukkan salah satu dari OID. Sensor ini mempunyai LED pada sisi masukan dan fototransistor pada sisi keluaran. Tegangan sumber dan tahanan seri menghasilkan arus melalui LED. Sebagai gantinya, cahaya dari LED mengenai fototransistor dan menyebabkan timbulnya arus balik.



Gambar 2.7. Gambar Skematik Dari OID

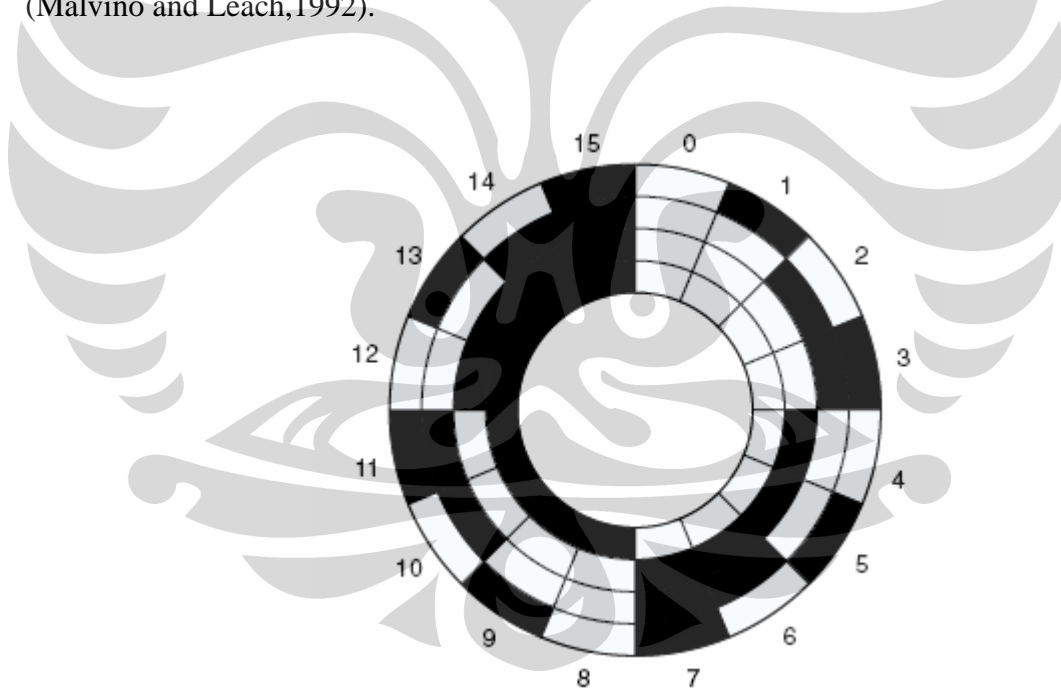
Tegangan keluaran tergantung pada arus balik yang dihasilkan. Bila tegangan masuk berubah maka jumlah cahaya pada LED juga akan berubah. Ini berarti tegangan keluar berubah sejalan dengan tegangan masuk. Itulah sebabnya mengapa gabungan antara LED dan fototransistor disebut optocoupler.

Keuntungan utama dari optocoupler adalah pemisahan secara listrik antara rangkaian masuk dan rangkaian keluarannya. Dengan optocoupler, hubungan yang ada antara masukan dan keluaran hanya berupa cahaya. Karena hal ini, kita dapat memperoleh resistansi penyekatan diantara dua rangkaian itu dalam ribuan megaohm. Pemisahan seperti ini kebetulan dibutuhkan dalam pemakaian – pemakaian bertegangan tinggi.

2.4.2. Sensor Shaft Encoder.

Penghitung atau **pencacah** (dlm bahasa Inggris: *counter*) adalah rangkaian sirkuit digital atau kadang-kadang berbentuk chip yang bisa dipakai untuk menghitung pulsa/sinyal digital (yang umumnya dihasilkan dari osilator). Penghitung ini bisa menghitung pulsa secara biner murni (binary counter), atau bisa menghitung secara desimal-terkodekan-secara-biner (decimal counter).

Pencacah digital adalah sebuah rangkaian yang mampu mencacah pulsa digital. Hasil cacahan berupa informasi digital dalam bentuk biner atau *binary coded decimal* (BCD). Pencacah biner yang umum, dirangkai dari IC7493, dimana setiap IC memiliki sebuah masukan, sebuah reset, dan keluaran 4 bit. Beberapa buah IC ini dapat disusun menjadi pencacah biner $4n$ bit, dimana n = banyaknya IC, dengan cara menghubungkan jalur bit tertinggi IC pertama menjadi masukan bagi IC berikutnya (Malvino and Leach,1992).

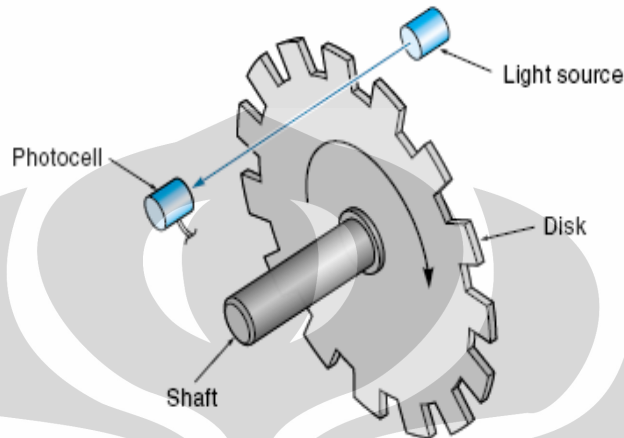


Gambar 2.8. Gambar piringan untuk mengetahui nilai biner jika berwarna gelap bernilai

1

Pada modul ini sensor ini berfungsi untuk mengkonversi gerak linier dan rotasi menjadi sinyal digital. Karena suatu encoder memiliki sejumlah masukan dan pada

saat tertentu hanya salah satu dari masukan-masukan itu yang berada pada keadaan 1, dan sebagai akibatnya suatu kode N-bit akan dihasilkan sesuai dengan masukan khusus yang telah di eksitasi. Untuk lebih jelasnya kita dapat lihat pada gambar 2.9



Gambar 2.9. Gambar Sensor Shaft Encoder

2.5. Prinsip Dasar PID

Sistem pengendali merupakan suatu sistem yang berfungsi untuk mengendalikan suatu sistem yang lain. Sistem pengendali di gunakan agar kinerja suatu sistem kendali menjadi lebih akurat dan efisien. Pada sistem pengendali di kenal beberapa istilah antara lain :

a. SP (Set Point)

Set Point adalah harga atau nilai dari keadaan yang ingin di capai pada proses.

b. Error

Error adalah nilai yang dihasilkan dari selisih antara Set Point dan Process Variabel.

c. MV (Manipulated Variable)

Manipulated Variable adalah harga atau nilai yang diatur agar proses menjadi stabil dan biasanya di gabungkan dengan input actuator (contohnya : control valve).

d. PV (Process Variable)

Process variable adalah sinyal hasil pemantauan terhadap proses atau plant. Process Variable umumnya adalah hasil pembacaan dari suatu pengendali (contohnya : thermocouple).

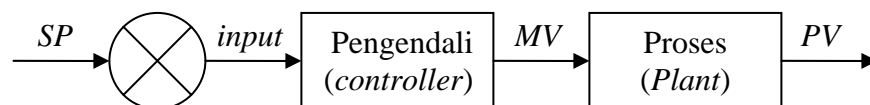
e. Plant

Plant adalah objek yang akan di kendalikan (contohnya : temperature) oleh suatu pengendali.

Secara umum sistem pengendalian terbagi atas dua jenis yaitu Open Loop Control Sistem dan Closed Loop Control Sistem. Untuk lebih jelasnya akan dibahas pada subbab dibawah ini.

2.5.1 Open Loop Control Sistem

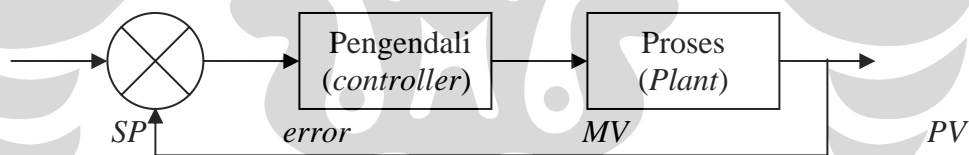
Open Loop Control Sistem atau sistem pengendali loop terbuka merupakan sistem pengendali dimana objek yang di control tidak di feed back ke pengendali sehingga pengendali hanya memberikan output jika di berikan suatu sinyal input. Pengendali jenis ini masih bersifat manual karena tidak akan terlepas dari intervensi atau campur tangan manusia. Pengendali ini tidak akan bekerja secara otomatis, karena adanya intervensi manusia dan hasil dari suatu proses yang di kendalikan tidak di bandingkan oleh pengendali itu sendiri. Gambar berikut menggambarkan sistem pengendali loop terbuka (*Open Loop Control Sistem*)



Gambar 2.9. Sistem PengendaliTerbuka

2.5.2. Closed Loop Control Sistem

Closed Loop Control Sistem atau sistem pengendali loop tertutup yaitu sistem pengendali dimana objek yang di control di *feed back* ke *input* pengendali. Input yang di berikan ke pengendali merupakan selisih antara besaran (PV) dan besaran (SP). Nilai selisih ini sering di sebut juga dengan *Error*. Tujuan dari pengendali adalah membuat nilai *Process Variable* (PV) sama dengan nilai *Set Point* (SP) atau nilai *Error* 0. Sinyal *Error* akan di olah oleh pengendali agar nilai (PV) sama dengan nilai (SP). Pengendali jenis ini bersifat otomatis karena objek yang akan di kendalikan di bandingkan lagi dengan input keadaan yang diinginkan, sehingga intervensi manusia dapat di hilangkan. Kinerja dari suatu pengendali ditentukan oleh semakin cepatnya respon pengendali untuk mengubah MV terhadap perubahan sinyal *error*, dan semakin kecilnya kesalahan yang terjadi. Gambar berikut menggambarkan sistem pengendali loop tertutup (*Closed Loop Control System*).



Gambar 2.10. Sistem Pengendali Tertutup

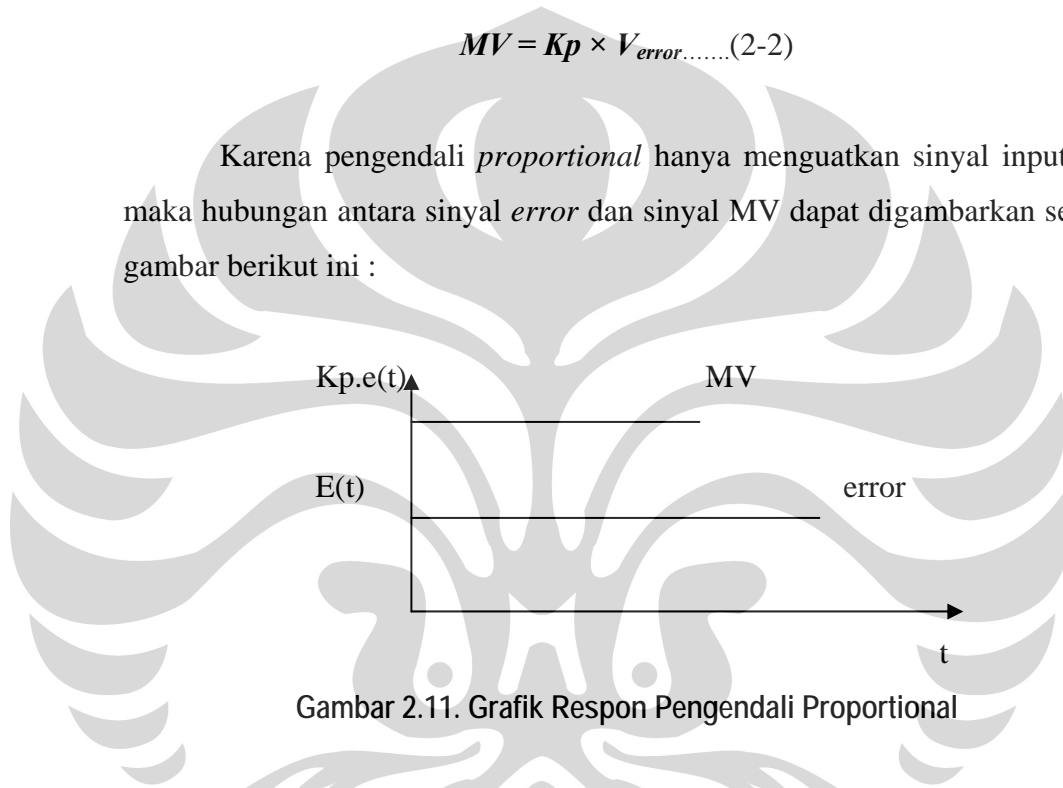
Pengendali P.I.D terdiri dari tiga macam pengendali yaitu pengendali Proportional (P), pengendali Intergral (I), dan pengendali Differensial (D). Masing-masing pengendali ini saling di kombinasikan sehingga didapatkan bentuk atau struktur dari P.I.D yaitu struktur paralel, struktur mix dan struktur seri.

2.5.2.1. Pengendali Proportional (P)

Pengendalian *proportional* berfungsi untuk mengalikan sinyal input dengan suatu besaran atau konstanta dengan nilai tertentu. Persamaan (2-2) adalah hubungan antara input (error) dan output (MV) pada pengendalian ini adalah :

$$MV = K_p \times V_{error} \dots (2-2)$$

Karena pengendali *proportional* hanya menguatkan sinyal input saja maka hubungan antara sinyal *error* dan sinyal MV dapat digambarkan seperti gambar berikut ini :



Gambar 2.11. Grafik Respon Pengendali Proportional

Ciri-ciri *controller* propotional harus diperhatikan ketika kontroler tersebut diterapkan pada suatu sistem. Secara eksperimen, pengguna controller proposional harus memperhatikan ketentuan-ketentuan berikut ini :

1. Kalau nilai K_p kecil, *controller* propotional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
2. Kalau nilai K_p dinaikkan, respon sistem menunjukkan semakin cepat mencapai keadaan stabilnya.

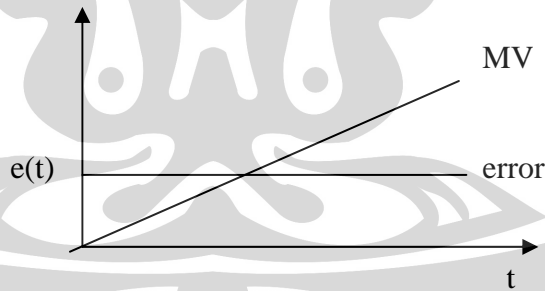
3. Namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil atau respon sistem akan berosilasi.

2.5.2.2 Pengendali Intergral (I)

Pengendali Integral berfungsi untuk mengintegrasikan sinyal input lalu di bagi dengan suatu besaran atau konstanta dengan nilai tertentu. Persamaan (2-3) adalah hubungan antara input (error) dan output (MV) pada pengendali ini adalah :

$$MV = Ti \times \int V_{error} dt \dots\dots(2-3)$$

Karena pengendali integral hanya mengintegrasikan sinyal input saja maka hubungan antara sinyal error dan sinyal MV dapat I gambarkan seperti grafik respon berikut ini :



Gambar 2.12. Grafik Respon Pengendalian Integral

Sinyal output *controller* integral merupakan luas bidang yang dibentuk oleh kurva kesalahan penggerak. Sinyal output akan berharga sama dengan harga sebelumnya ketika sinyal bernilai nol. (Ogata, 1997, 236) menunjukkan contoh sinyal kesalahan yang disulutkan ke dalam *controller* integral dan output *controller* integral terhadap perubahan sinyal kesalahan tersebut.

Ketika sinyal kesalahan berlipat ganda, maka nilai laju perubahan output *controller* berubah menjadi dua kali dari semula. Jika nilai konstanta

integator berubah menjadi lebih besar, sinyal kesalahan yang relatif kecil dapat mengakibatkan laju output menjdai besar (Johnson,1993,375)

Ketika digunakan, *controller* integral mempunyai karakteristik sebagai berikut :

1. Output *controller* membutuhkan selang waktu tertentu, sehingga *controller* integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, output *controller* akan bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, output akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i .
4. Konstanta integral K_i yang berharga besar akan mempercepat hilangnya offset. Tetapi semakin besar nilai konstanta K_i , akan mengakibatkan peningkatan osilasi dari sinyal output *controller*.

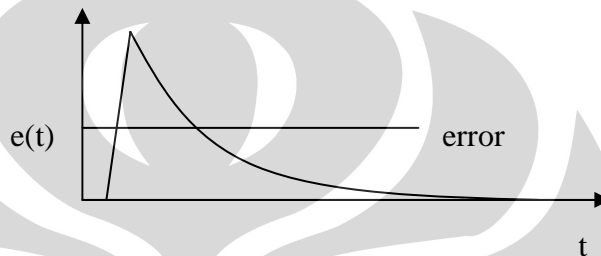
Berdasarkan karakteristik *controller* tersebut, *controller* diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja *controller* diferensial hanyalah efektif pada lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler diferensial tidak pernah digunakan tanpa ada *controller* lain dalam sebuah sistem.

2.5.2.3 Pengendali Differensial (D)

Pengendalian differensial berfungsi untuk mengdeferensialkan sinyal input lalu dikalikan dengan suatu besaran atau konstanta dengan nilai tertentu. Persamaan (2-4) adalah hubungan antara input (error) dan output (MV) pada pengendalian ini adalah :

$$MV = \frac{Td \times d V_{error}}{Dt} \dots\dots(2-4)$$

Karena pengendalian differensial hanya mengdifferensialkan sinyal input saja, maka hubungan antara sinyal error dan sinyal MV dapat digambarkan seperti grafik respon berikut ini :



Gambar 2.13. Grafik Respon Pengendali Differensial

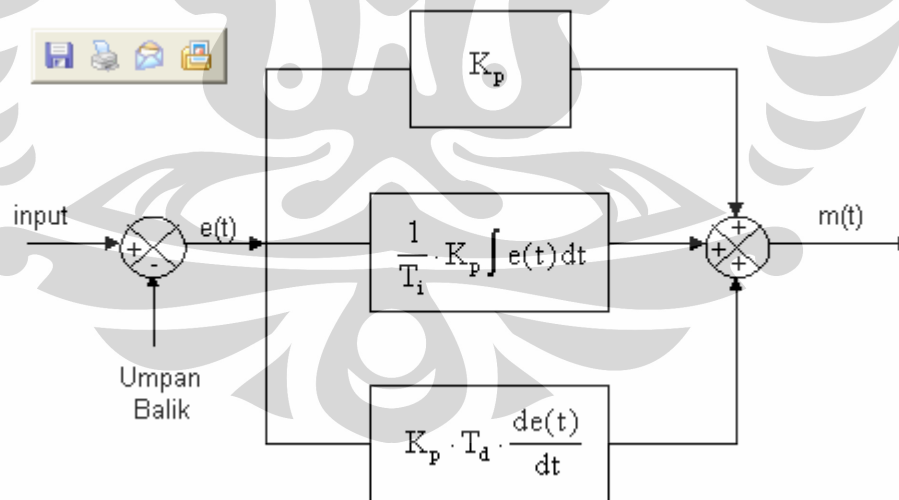
Karakteristik controller diferensial adalah sebagai berikut :

1. *Controller* ini tidak dapat menghasilkan output bila tidak ada perubahan pada masukannya (Berupa sinyal kesalahan).
2. Jika sinyal kesalahan berubah terhadap waktu, maka output yang dihasilkan *controller* tergantung pada nilai T_d dan laju perubahan sinyal kesalahan.
3. *Controller* diferensial mempunyai suatu karakter untuk mendahului, sehingga *controller* ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi *controller* diferensial dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif, dan cenderung meningkatkan stabilitas sistem (Ogata,1997,240)

2.5.2.4 Pengendali PID

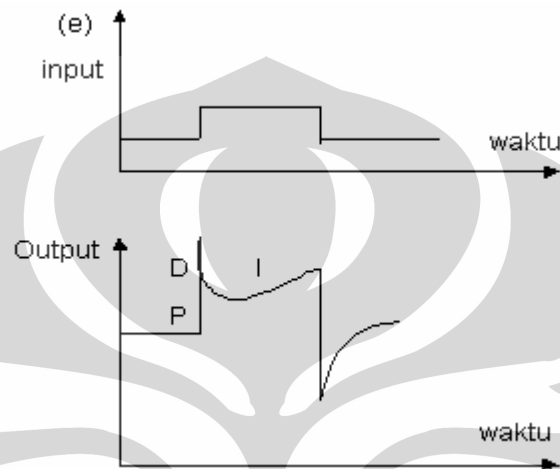
Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proposional plus integral plus diferensial (PID). Keuntungan lainnya adalah kemampuan merespon perubahan yang cepat. Perubahan ini diakibatkan gangguan (*disturbance*). Output proses atau proses variable (PV) akan berubah nilainya bila mendapatkan gangguan dari luar ataupun perubahan nilai dari set point. Bila perubahan yang terjadi diakibatkan oleh nilai dari set point, maka nilai PV akan mengikuti nilai dari set point tersebut.

Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan offset dan menghasilkan perubahan awal yang besar (Gunterus,1994,8-10).



Gambar 2.14. Blok diagram kontroler PID analog

Keluaran controller PID merupakan jumlahan dari keluaran kontroler proporsional, keluaran kontroler integral. Gambardi bawah ini menunjukkan hubungan tersebut.



Gambar 2.15. Grafik hubungan dalam fungsi waktu antara sinyal keluaran dengan masukan untuk controller PID

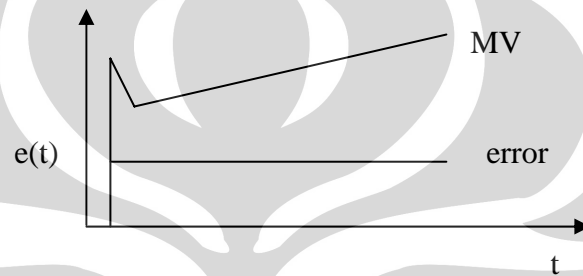
Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta K_p , T_i , dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

2.5.2.5 Pengendalian P.I.D Struktur Paralel

Pengendali P.I.D dengan struktur parallel merupakan sebuah pengendali yang terdiri dari kombinasi pengendali proporsional, pengendali differensial dan pengendali integral yang disusun parallel. Masing-masing

input dari pengendali proportional, pengendali differensial dan pengendali integral dihubungkan ke satu titik dan di hubungkan ke sinyal error.

Karena output pengendali P.I.D struktur parallel merupakan jumlah dari sinyal output pengendali proportional, pengendali differensial dan pengendali integral, maka sinyal MV dapat digambarkan seperti grafik respon berikut ini :

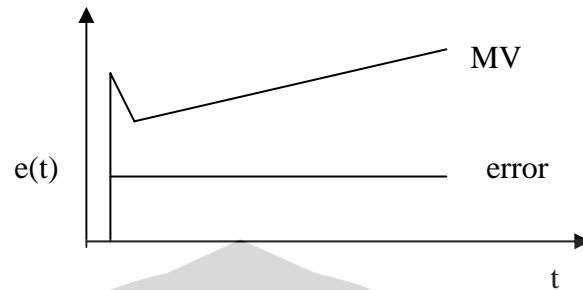


Gambar 2.16. Grafik Respon Pengendali P.I.D Struktur Paralel

2.5.2.6 Pengendalian P.I.D Struktur Mix

Pengendali P.I.D dengan struktur mix merupakan sebuah pengendali yang terdiri dari kombinasi pengendali proportional, pengendali differensial, dan pengendali integral yang disusun secara seri dan parallel. Input pengendali proportional disusun secara seri dengan sinyal error, sedangkan pengendali differensial dan pengendali integral disusun secara parallel dan kedua inputnya dihubungkan ke satu titik dan di hubungkan ke output pengendali proportional.

Karena output pengendali P.I.D struktur mix merupakan jumlah dari sinyal output pengendali proportional, pengendali differensial, dan pengendali integral, maka sinyal MV dapat digambarkan seperti grafik respon berikut ini

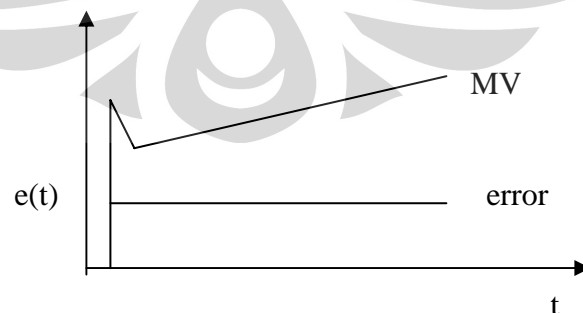


Gambar 2.17. Grafik Respon Pengendali P.I.D Struktur Mix

2.5.2.7 Pengendalian P.I.D Struktur Seri

Pengendali P.I.D dengan struktur seri merupakan sebuah pengendali yang terdiri dari kombinasi pengendali proporsional, pengendali differensial dan pengendali integral yang disusun seri. Input pengendali proporsional dihubungkan dengan sinyal error. Input pengendali integral dihubungkan dengan output dari pengendali proporsional. Dan input pengendali differensial dihubungkan dengan output dari pengendali integral.

Karena output pengendali P.I.D struktur seri merupakan jumlah dari sinyal output pengendali proporsional, pengendali differensial, dan pengendali integral maka MV dapat di gambarkan seperti grafik respon berikut ini



Gambar 2.18. Grafik Respon Pengendali P.I.D Struktur Seri

BAB 3

ANALISA RANGKAIAN DAN PROGRAM

Pada modul pengendali posisi ini sangat diperlukan rangkaian interface untuk menghubungkan antara hardware dan software. Rangkaian interface ini juga digunakan sebagai pengirim data antara komputer dan rangkaian PWM. Dan proses program mulai dari awal serta analisisnya hingga didapatkan data yang diperlukan.

3.1. Rangkaian Interface

Rangkaian *interface* ini digunakan untuk mengirim atau menerima data dari komputer ke rangkaian PWM atau sebaliknya. Rangkaian ini terdiri dari beberapa komponen yaitu DB 25 sebagai port paralel yang menghubungkan komputer dengan rangkaian ini, DB 15 digunakan untuk menghubungkan antara mekanik dan rangkaian interface, Ic 74LS245 sebagai transmitter bus, Ic 74LS244 digunakan sebagai buffer atau line receiver, Ic 74LS139 digunakan sebagai Demultiplexer, Ic 74LS04 sebagai inverting dan komponen yang terakhir yaitu kapasitor digunakan untuk buffer tegangan pada Ic.

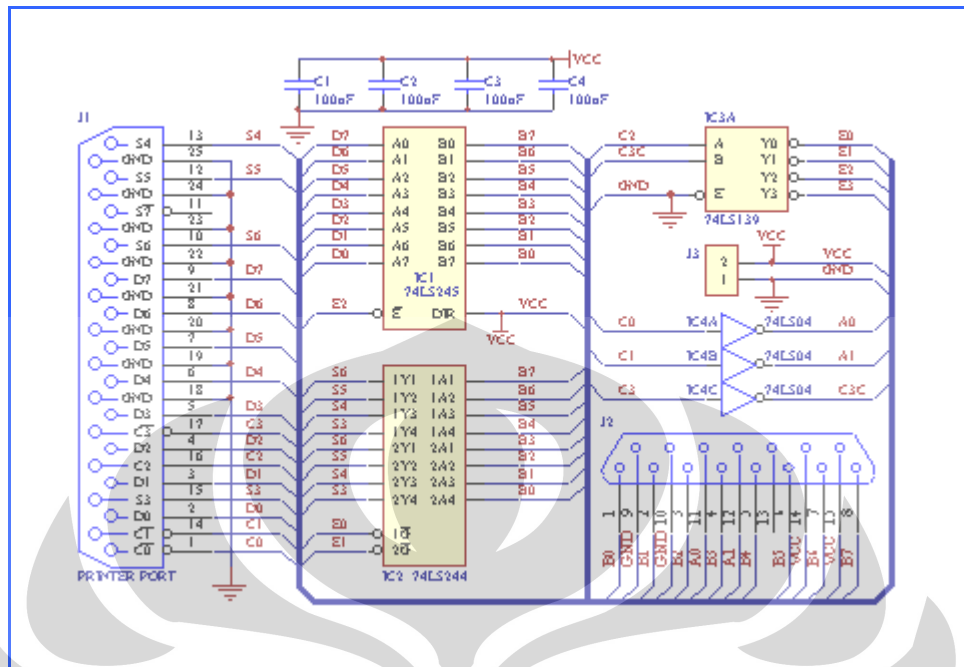
Disain rangkaian ini didasarkan pada prinsip kerja alat yang akan dibuat yaitu sebagai berikut: sinyal dari pulsa-pulsa yang dihasilkan oleh motor DC diubah dalam bentuk besaran biner dengan menggunakan sensor *shaftencoder*. Dengan menggunakan rangkaian pencacah dan rangkaian PWM. Perubahan nilai pulsa yang dihasilkan akan menjadikan perubahan PWM keluaran dari rangkaian ini berupa bilangan biner antara 0 – 255. Informasi ini sebagai pengolahan lebih lanjut dari parameter sinyal PWM itu kemudian disalurkan melalui sebuah transmisi kabel tunggal secara bergantian dengan menggunakan rangkaian multiplexer. Dua buah

informasi dalam satu saluran ini kemudian di demultiplexer oleh bagian penerima di ujung kabel pada tempat yang jauh.

Sinkronisasi multiplekser dan demultiplekser dikendalikan oleh komputer melalui *interface*. Setelah didemultiplekser, sinyal PWM tersebut diolah sehingga membentuk sinyal blok yaitu suatu gelombang persegi atau digital yang akan dicacah oleh suatu rangkaian pencacah. Pencacahan sinyal digital ini pewaktuannya dikendalikan pula oleh komputer melalui *interface*. Hasil cacahan berupa nilai dalam biner 24 bit dihubungkan dengan port masukan pada *interface* yang kemudian dibaca oleh komputer sebagai suatu nilai biner dalam rentang 0 sampai 255. Pengaturan *interface* dalam mengendalikan multiplekser, pewaktuan pencacah dan pembacaan hasil cacahan sepenuhnya diatur oleh program komputer yang dibuat dalam bahasa Visual Basic.

Unit komputer digunakan untuk mengendalikan peralatan melalui *interface* yang dipasang pada slot ekspansinya. Untuk mengaktifkan *interface* dalam fungsinya sebagai perantara komputer dengan modul pengendali posisi diperlukan suatu program pengendali. Program dibuat dalam bahasa Basic menggunakan Visual Basic. Bahasa ini memungkinkan pengaksesan perangkat keras dari komputer.

Keuntungan dari rangkaian interface dengan menggunakan port parallel adalah pengiriman datanya lebih cepat dan kendala yang dihadapi adalah jalur untuk pengiriman data sangat banyak sehingga tidak efektif untuk pemasangan kabel.



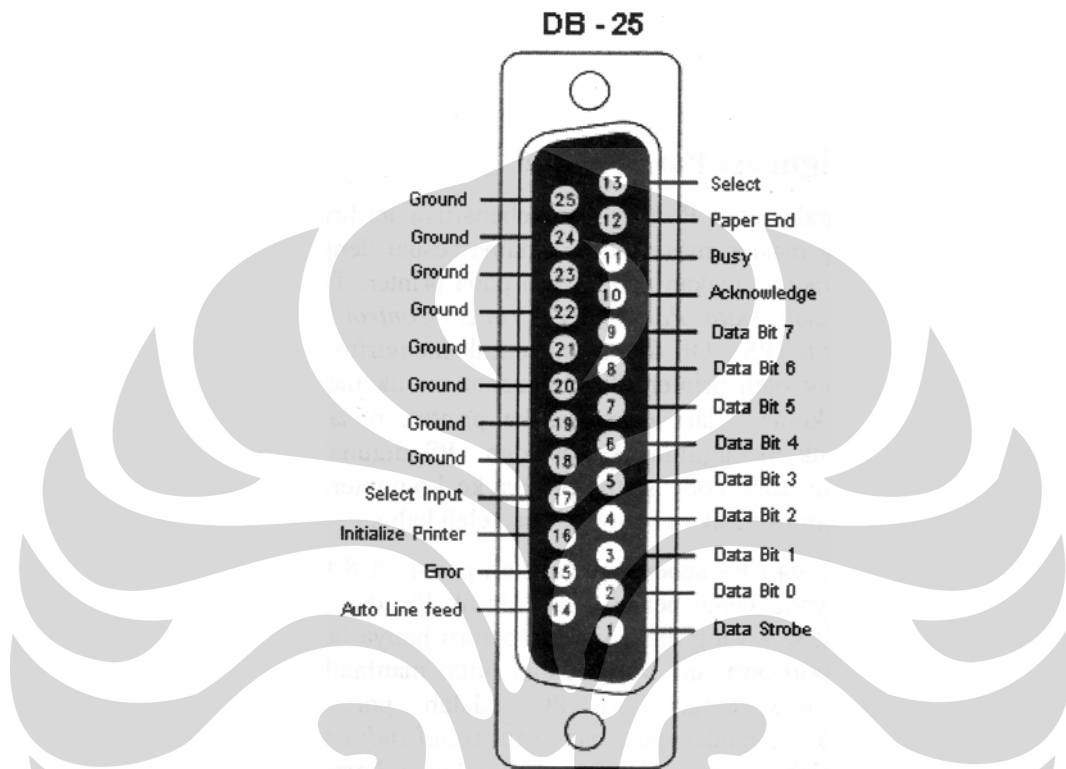
Gambar 3.1. Skematik rangkaian Interface.

3.2. Konfigurasi Port Paralel

Port parallel atau disebut juga port printer terdiri dari tiga bagian yang masing-masing diberi nama sesuai dengan tugasnya. Tiga bagian tersebut adalah *Data Port (DP)*, *Printer Kontrol (PC)*, dan *Printer Status (PS)*. DP digunakan untuk mengirim data, PC digunakan untuk mengirim kode-kode dari komputer, dan PS digunakan untuk mengirimkan kode-kode status printer ke komputer.

DP, PC, dan PS sebenarnya adalah port-port 8 bit, namun yang benar-benar 8 bit hanya DP. Untuk PC dan PS, hanya beberapa bit saja yang dipakai yang berarti hanya beberapa bit saja yang digunakan untuk keperluan *interfacing*. Port PC adalah port baca/tulis (*read/write*), PS adalah port baca saja (*read only*), sedangkan port DP adalah port baca/tulis juga. Akan tetapi, kemampuan ini hanya dimiliki oleh *Enhance Paralel Port (EPP)*. Sedangkan untuk port paralel yang standar hanya memiliki kemampuan untuk tulis saja. Pada EPP pengaturan jalur data DP dilakukan lewat bit 5 PC. Jika bit 5 PC bernilai 0, maka jalur data dwi-arah DP menjadi output

dari port parallel, sebaliknya jika bit PC bernilai 1, maka jalur data dwi-arah DP menjadi input dari port parallel.



Gambar 3.2. Skematik port paralel.

Selengkapnya konfigurasi slot DB-25 female yang terdapat pada belakang komputer dapat dilihat pada gambar 3.2 dan konfigurasi dari DP, PC, dan PS dapat dilihat pada tabel 3.1. Pin-pin dengan keterangan komplemen akan berlogika tinggi pada keadaan awal.

Tabel 3.1. Tabel Konfigurasi pin dan nama sinyal konektor paralel standar DB- 25.

Nomor Pin	Nama Sinyal	Direction	Register	Komplemen
1	Strobe	In/Out	Kontrol bit 0	Ya
2	Data 0	Out	Data Bit 0	
3	Data 1	Out	Data Bit 1	
4	Data 2	Out	Data Bit 2	
5	Data 3	Out	Data Bit 3	
6	Data 4	Out	Data Bit 4	
7	Data 5	Out	Data Bit 5	
8	Data 6	Out	Data Bit 6	
9	Data 7	Out	Data Bit 7	
10	Ack	In	Status Bit 6	
11	Busy	In	Status Bit 7	Ya
12	Paper Out	In	Status Bit 5	
13	Select	In	Status Bit 4	
14	Auto-Linefeed	In/Out	Kontrol Bit 1	Ya
15	Error / Fault	In	Status Bit 5	
16	Initialize	In/Out	Kontrol Bit 2	
17	Select In	In/Out	Kontrol Bit 3	Ya
18 – 25	Ground	Gnd		

Untuk dapat menggunakan port paralel, kita harus mengetahui alamatnya. *Base address* LPT1 biasanya adalah 888 (378h) dan biasanya 632 (278h). Alamat tersebut adalah alamat yang umumnya digunakan, tergantung dari jenis komputer. Setelah mengetahui alamat dari port paralel, maka kita dapat menentukan alamat DP, PC dan PS. Alamat DP adalah *base address* dari port paralel tersebut. Untuk alamat

PS adalah base address + 1, dan alamat PC adalah base address + 2. Tabel 3.2 adalah tabel alamat masing-masing port yang umumnya digunakan.

Tabel 3.2. Tabel alamat masing-masing port.

Nama port	Alamat register
LPT1 DP	378h/888
LPT1 PS	379h/889
LPT1 PC	37Ah/890

Pada program yang digunakan pada modul ini yang digunakan sebagai penunjuk alamat adalah

```
'Deklarasi Alamat input dan output
```

```
Const OutData = &H378
```

```
Const InData = &H379
```

```
Const Kontrol = &H37A
```

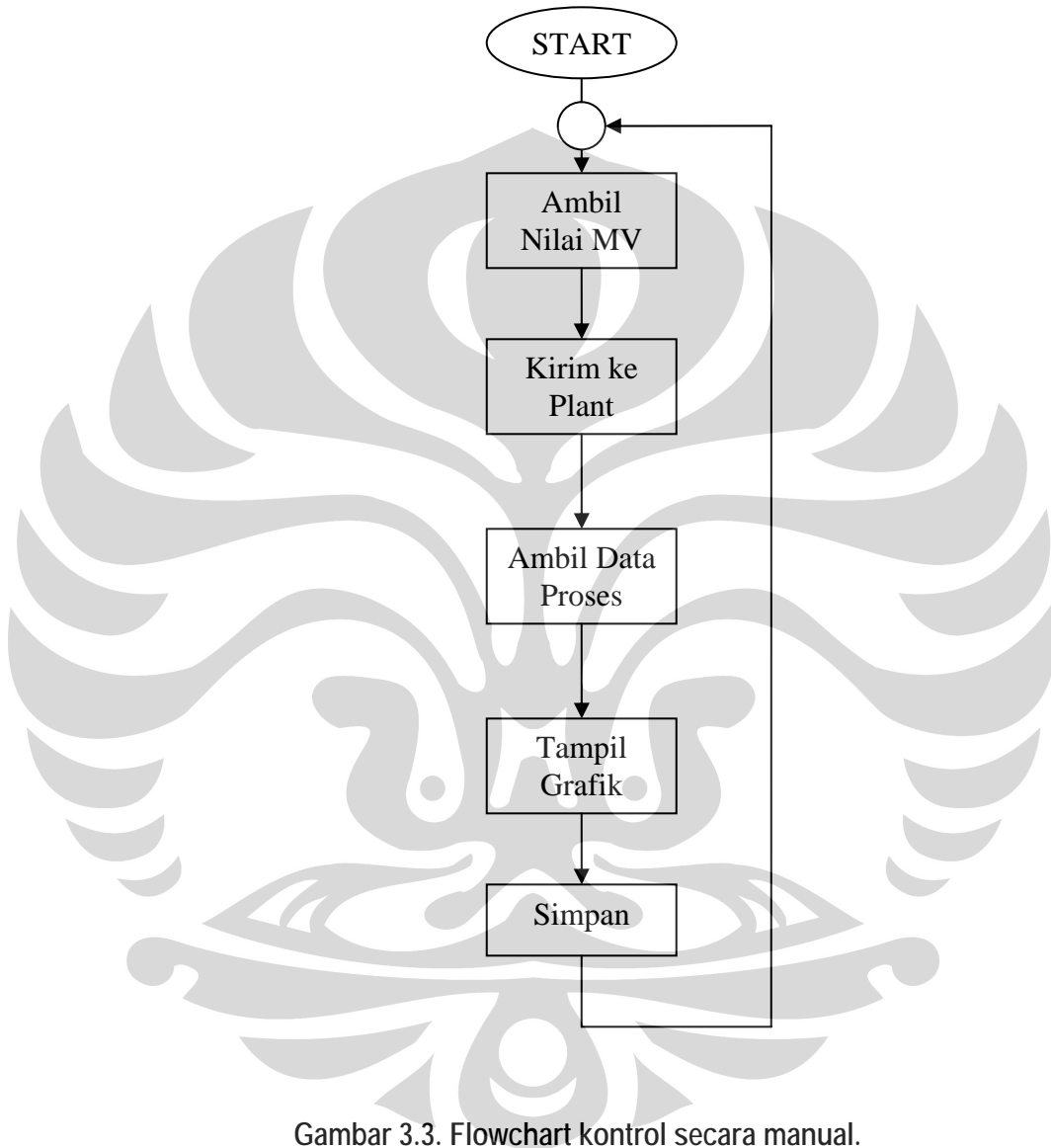
3.3. Analisa Program

Pada modul ini dibutuhkan pengendalian yang bersifat digital agar memudahkan proses pengendalian. Untuk itu dibutuhkan sebuah program pengendali dan penulis menggunakan bahasa Visual Basic sebagai bahasa pengendalian. Dalam proses pengendalian ini menggunakan dua kontrol yaitu secara manual dan Auto. Untuk lebih jelasnya kita dapat melihat pada subbab dibawah ini

3.3. 1. Kontrol Secara Manual.

Saat kita melakukan proses kendali kita akan dihadapkan dengan dua option yaitu manual dan auto. Saat kita memilih proses kontrol dengan menggunakan kontrol secara manual maka program akan meminta nilai MV terlebih dahulu pada PID dengan batasan nilainya 0 sampai 100. Nilai MV yang dimasukkan akan masuk ke plant untuk mendapatkan nilai PV. Nilai PV yang didapatkan akan tampil dalam

bentuk angka pada from data dan akan tampil dalam bentuk grafik juga. Setelah itu nilai akan di simpan dan program akan mengambil data kembali dari plant. Program ini akan terus berlangsung sampai kita menghentikan program.

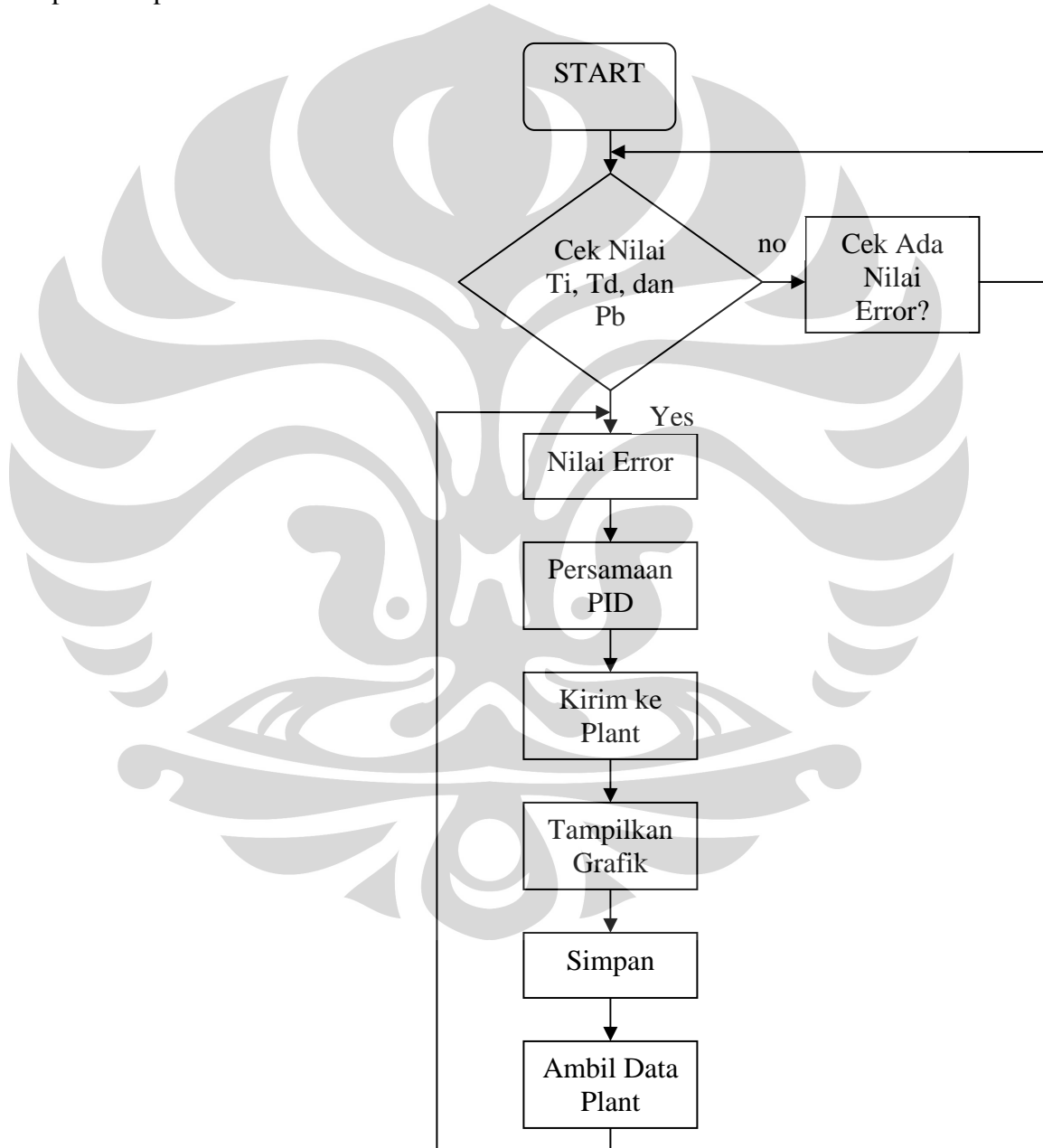


Gambar 3.3. Flowchart kontrol secara manual.

3.3.2. Kontrol Secara Auto.

Jika program control yang digunakan adalah mode auto maka kita harus memilih struktur PID yang digunakan secara parallel, seri dan mix. Setelah itu masukan nilai yang dibutuhkan oleh PID yaitu TI, TD, SP, dan PB. Dari data itu akan

didapatkan nilai error dengan rumus $SP - PV$. Lalu semua data dimasukkan ke dalam persamaan PID nya sesuai dengan struktur yang dipilih. Data kemudian akan masuk ke plant. Dari data itu didapatkan nilai MV dan PV yang akan tampil pada from data dan dalam bentuk grafik juga. Dan program akan mengambil data pada plant. Program akan terus berlanjut sampai program dihentikan. Untuk lebih jelasnya kita dapat lihat pada flowchart dibawah ini :



Gambar 3.4. Flowchart kontrol secara Auto.

BAB 4

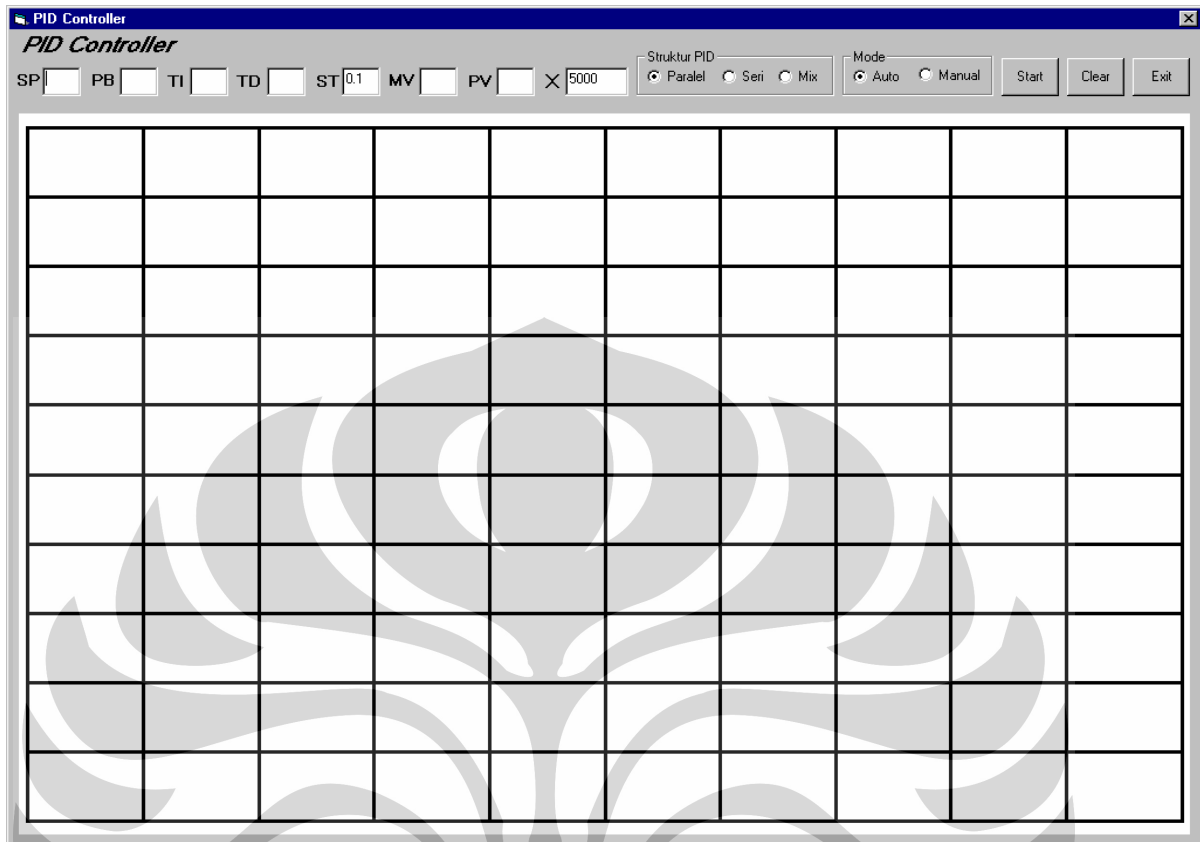
HASIL PENELITIAN DAN ANALISA

Setelah melakukan perancangan sistem pengendali kita dapat melihat hasil dari penelitian ini, dari hasil ini kita dapat menentukan apakah data yang dihasilkan sangat baik atau jauh dari teori yang kita gunakan. Untuk lebih jelasnya kita dapat lihat pada subbab dibawah ini :

4.1. Cara Pengambilan Data

Dalam proses pengambilan data kita menggunakan program dengan bahasa Visual Basic. Dengan tampilan form kontrol seperti gambar 4.1 dibawah, untuk menjalankan program kita harus mengisi optionnya terlebih dahulu jika kita ingin menggunakan mode manual kita harus mengisi nilai MV terlebih dahulu. Dan jika secara auto kita harus mengisi SP, PB, TI dan TD, serta mode ini juga harus memilih struktur PID yang digunakan Paralel, Mix, dan Seri. Nilai X dapat juga diubah-ubah jika kita ingin mendapatkan nilai respon yang lebih jelas lagi. Sama seperti nilai X, nilai ST juga dapat diubah untuk lamanya waktu saat pengambilan data tapi dalam program sudah diset selama 0.1 ms.

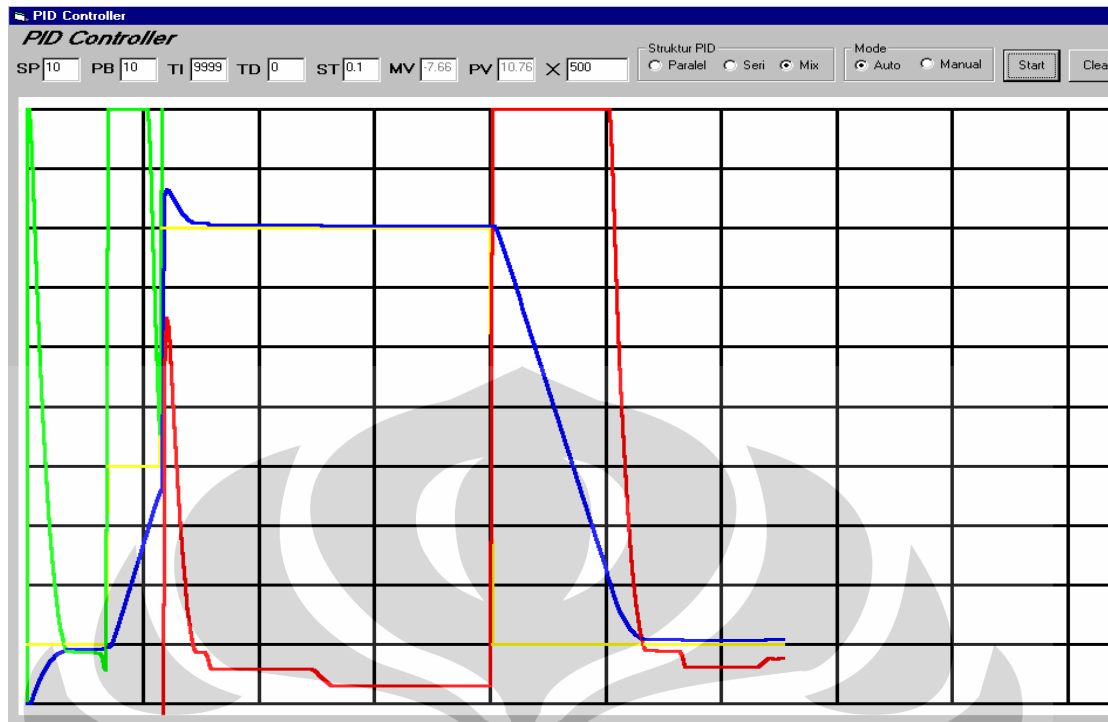
Saat program sudah akan distart pastikan semua nilai sudah terisi. Jika sudah maka akan terlihat pada form gambar respon yang dihasilkan dalam bentuk grafik. Dan untuk melihat data dalam bentuk angka kita dapat melihat dalam bentuk Excel.



Gambar 4.1. Form Data Pada Program.

4.2. Hasil Penelitian.

Saat pengambilan data yang kita gunakan hanya mode auto saja, karena mode manual hanya digunakan untuk maju dan mundur saja. Untuk itu saat menggunakan mode auto nilai yang diisi yaitu $P_b = 10$, $T_i = 9999$, $T_d = 0$, nilai S_p bergantian yaitu 10,40,80 dan kembali ke10. dan struktur PID yang dipakai adalah Mix, struktur ini digunakan agar kita mendapatkan respon yang baik. Respon yang dihasilkan adalah sebagai berikut :

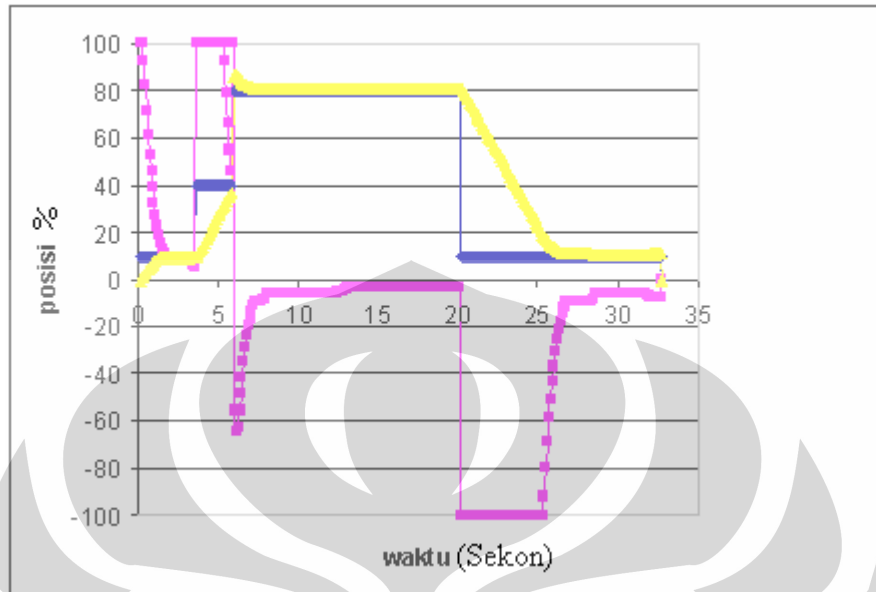


Gambar 4.2. Respon yang dihasilkan pd Pb 10.

Keterangan Gambar :

- ❖ Warna kuning : Set Point
- ❖ Warna biru : PV (Proses Variable)
- ❖ Warna Hijau dan Merah : MV (Manipulated Variable)

Gambar diatas menunjukkan respon sistem yang didapatkan masih kurang baik karena masih terdapat nilai overshoot, nilai overshoot yaitu nilai PV melebihi dari nilai set Point yang diberikan. Sehingga motor akan bergerak mundur lagi karena nilai yang dituju terlewat. Ini yang menandakan kurang bagusnya respon sistem yang didapatkan. Dari gambar respon sistem diatas kita dapatkan hasil gambar dari nilai MV, PV, dan SP, gambar ini berdasarkan dari data yang diperoleh excel.



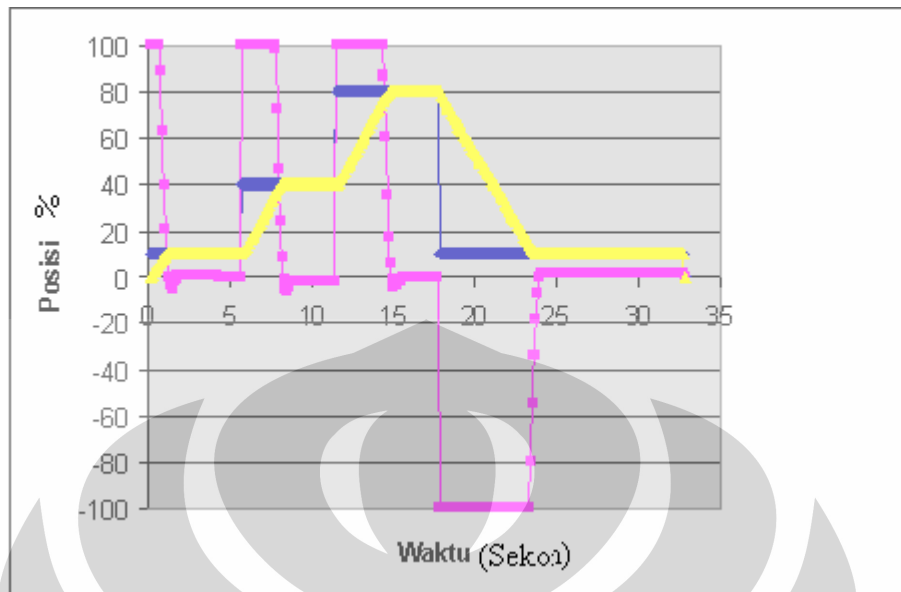
Gambar 4.3. Respon yang dihasilkan pd Pb 10 dgn data excel.

Keterangan Gambar :

- warna merah muda = MV
- Warna Kuning = PV
- Warna Biru = Set Point

Data yang diperoleh oleh excel juga menyatakan adanya nilai overshoot pada respon dengan menggunakan nilai Pb sama dengan 10. Gambar ini tidak berbeda dengan dengan gambar yang dihasilkan dari form data diatas hanya nilai overshoot nya tidak terlalu jelas.

Untuk lebih meyakinkan lagi kalau penelitian yang kita lakukan mendapatkan respon sistem yang hampir mendekati nilai MV dan PV sama, maka kita melakukan percobaan lagi dengan nilai Pb yang berbeda-beda, yaitu Pb = 5 tetapi nilai Ti dan Td sama. Nilai respon yang diperoleh akan terlihat pada gambar dibawah ini:



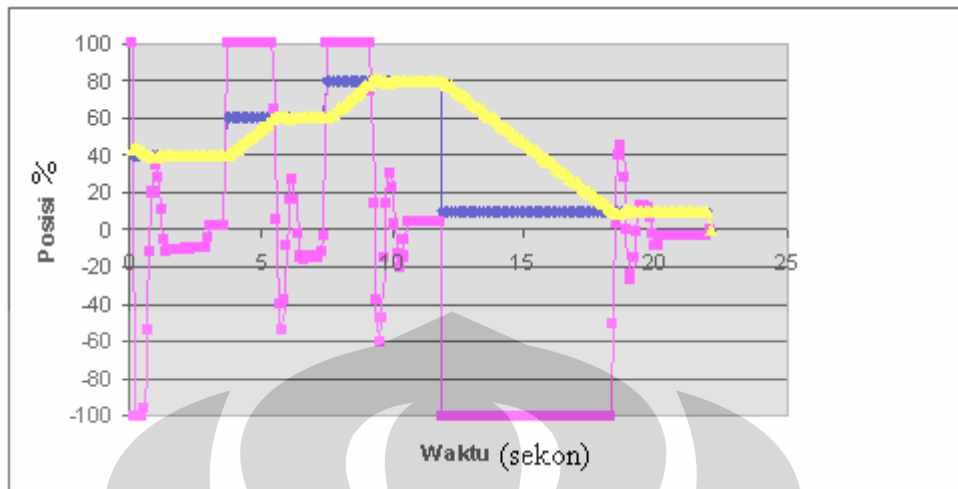
Gambar 4.4. Respon yang dihasilkan pd Pb 5 dgn data excel.

Keterangan Gambar :

- warna merah muda = MV
- Warna Kuning = PV
- Warna Biru = Set Point

Dari gambar yang dihasilkan dapat dilihat nilai responnya sangat baik sekali karena nilai overshootnya sama sekali tidak ada, ini menunjukkan nilai MV dan PV hampir sam. Dengan nilai Pb sama dengan 5, maka respon sistem menunjukkan semakin cepatnya mencapai kestabilan.

Percobaan dengan menggunakan nilai Pb yang berbeda lagi, yaitu $Pb = 2$, maka respon yang dihasilkan dapat kita lihat pada gambar dibawah ini:



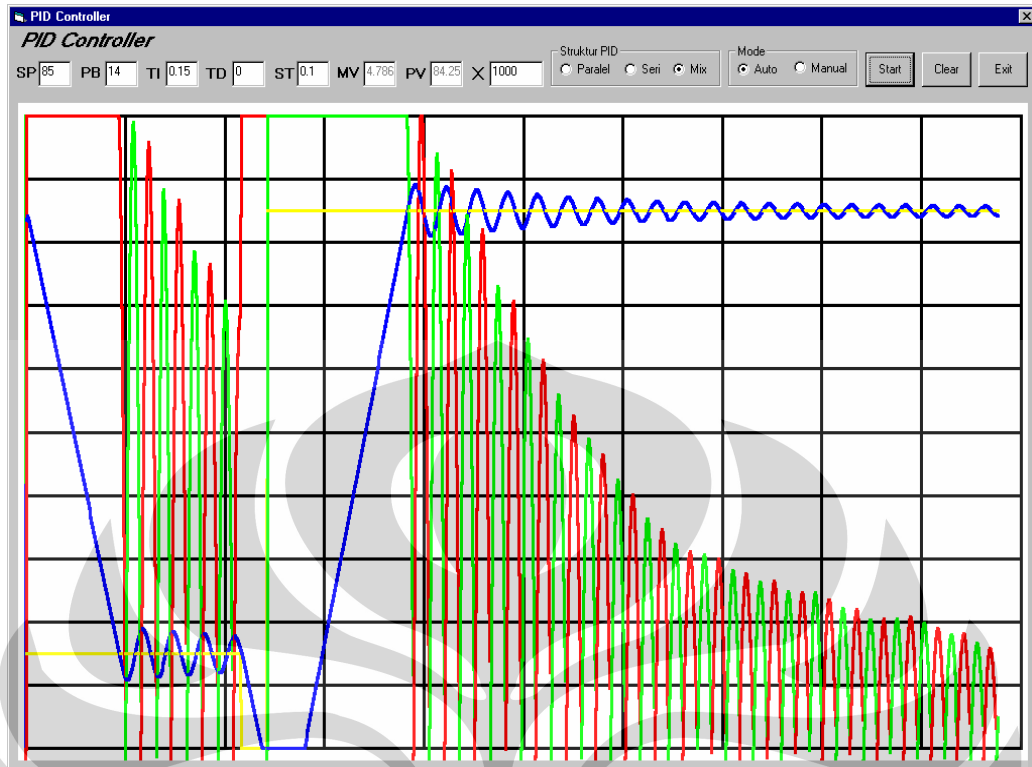
Gambar 4.5. Respon yang dihasilkan pd Pb 2 dgn data excel.

Keterangan Gambar :

- warna merah muda = MV
- Warna Kuning = PV
- Warna Biru = Set Point

Dari gambar diatas kita dapat melihat respon sistem yang didapatkan hampir sama dengan respon sistem yang menggunakan nilai $P_b = 5$. Respon sistem yang dihasilkan sangat bagus.

Kita mengambil data yang berbeda dengan sebelumnya maka didapatkan respon yang hampir sama. Nilai yang kita masukkan adalah $P_b = 14$, $T_i = 0.2$, $T_d = 0$, dan struktur PID nya masih menggunakan Mix. Gambar respon yang dihasilkan

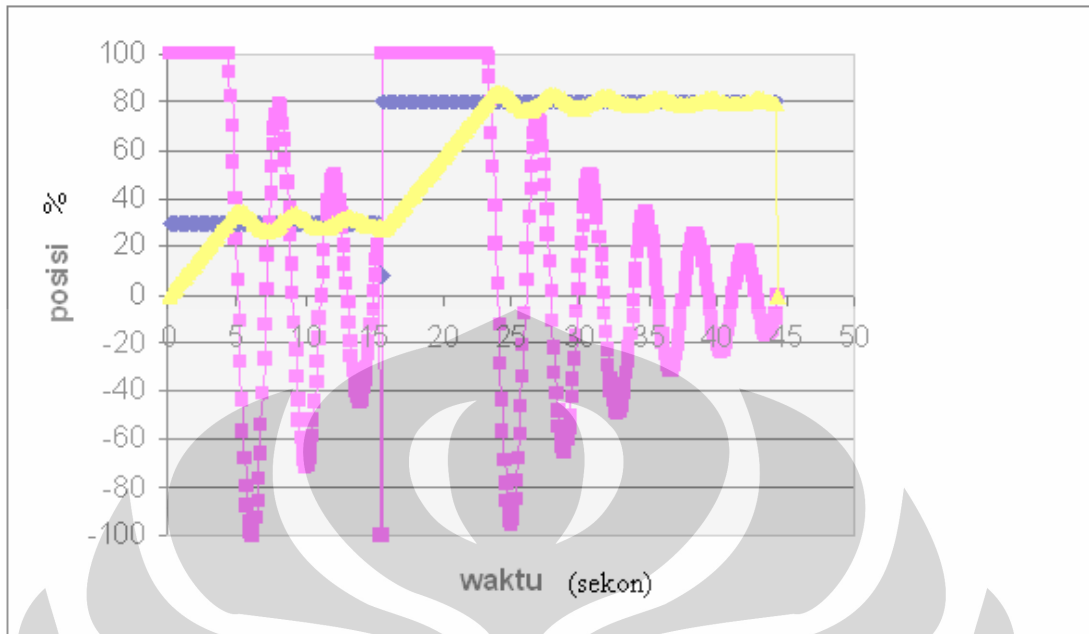


Gambar 4.6. Respon yang dihasilkan pd Pb14.

Keterangan Gambar :

- ❖ Warna kuning : Set Point
- ❖ Warna biru : PV (Proses Variable)
- ❖ Warna Hijau dan Merah : MV (Manipulated Variable)

Dari gambar respon sistem yang diperoleh sangat jelek karena selain ada nilai overshootnya nilai PV nya juga tidak stabil karena terus berosilasi. Kalau dalam mekaniknya motor DC akan bergerak maju mundur terus menerus, sehingga nilai yang akan dituju tidak terbaca.



Gambar 4.7. Respon yang dihasilkan pd Pb14 dgn excel.

Keterangan Gambar :

- warna merah muda = MV
- Warna Kuning = PV
- Warna Biru = Set Point

Dalam pengambilan data kita menggunakan Trial Error yang pada dasarnya mencari nilai kritis yaitu respon yang masih dapat berhasil. Untuk mendapatkannya kita harus menghilangkan nilai integral dan differential, Kc yang digunakan dimulai dari nilai yang paling kecil dan perlahan-lahan dinaikkan sampai respon menghasilkan osilasi (nilai Kc didapatkan dari $100/P_b$) dan nilai proposionalnya sama dengan $\frac{1}{2}$ dari nilai osilasi yang dihasilkan.

Tabel 4.1. Ziegler – Nichols Original.

Parameter	Kc	I	P
P	0.5Kcu	–	–
PI	0.45Kcu	Pu/1.2	–
PID	0.6 Kcu	Pu/2	Pu/8

Saat $P_b = 10$, maka perhitungan nilai Kcu nya adalah :

- $K_{cu} = 100 / P_b$
- $K_{cu} = 100/10$
- $K_{cu} = 10$

Dari respon yang dihasilkan maka didapatkan nilai Pu, nilai Pu yaitu nilai dari puncak ke puncak. Nilainya adalah $6.5 - 1.5 = 4$. Kemudian semua nilai yang ada kita masukkan kedalam persamaan yang berada di tabel :

❖ Pada parameter P

$$K_c = 0.5 (K_{cu})$$

$$K_c = 0.5 (10)$$

$$K_c = 5$$

❖ Pada parameter PI

$$K_c = 0.45 (K_{cu})$$

$$T_i = P_u / 1.2$$

$$K_c = 0.45 (10)$$

$$T_i = 4 / 1.2$$

$$K_c = 4.5$$

$$T_i = 3.333$$

❖ Pada parameter PID

$$K_c = 0.6 (K_{cu})$$

$$T_i = P_u / 2$$

$$P = P_u / 8$$

$$K_c = 0.6 (10)$$

$$T_i = 4 / 2$$

$$P = 4 / 8$$

$$K_c = 6$$

$$T_i = 2$$

$$P = 0.5$$

Saat menggunakan parameter proposional hasil dari respon yang didapatkan sangat baik sekali, namun menurut teori yang digunakan seharusnya ada nilai error. Nilai error ini didapatkan dari selisih antara SP (Set Point) dengan PV (Proses

Variable). Dan dari data yang dihasilkan pada excel error yang terjadi ada walaupun hanya kecil sekali. Untuk lebih jelasnya kita dapat lihat pada tabel dibawah ini :

Tabel 4.2. Hasil Respin dengan nilai Pb = 10 dgn Excel.

I	SP	MV	PV
7.0	80	-16.79177	81.398
7.1	80	-13.97929	81.143
7.2	80	-11.42721	80.933
7.3	80	-9.33087	80.867
7.4	80	-8.6668	80.865
7.5	80	-8.6538	80.864
7.6	80	-8.6407	80.863
7.7	80	-8.62277	80.8606
7.8	80	-8.6017	80.7903
7.9	80	-7.8986	80.617

Tabel diatas hanya 10 respon yang terjadi dari 330 data yang tersimpan pada excel. Kita dapat lihat perbandingan nilai Set Point dengan nilai PV sangat kecil sekali perbedaannya, sehingga pada grafik yang ditampilkan respon terlihat sangat bagus.

BAB 5

KESIMPULAN DAN SARAN

5.1.Kesimpulan

Setelah kita melakukan perancangan sistem serta pengujian terhadap pengendalian sistem tersebut, maka penulis dapat mengambil kesimpulan bahwa :

1. Data yang dihasilkan berupa nilai MV dan PV. Nilai - nilai ini akan bergantung dari nilai set point yang berikan ke dalam sistem pengendalian ini
2. Parameter pengontro PID berupa Kc, Ti dan Td ditentukan melalui metode osilasi. Dari metode itu didapatkan nilai Kcu dan Pu yang kemudian digunakan untuk menghitung dengan metode Ziegler – Nichols.
3. Nilai PB, TI dan TD sangat berpengaruh pada respon saat kita memilih mode auto dan melakukan proses pengendalian secara PID.

5.2.Saran

1. Metode yang digunakan saat pengambilan data tidak terlalu efektif sehingga respon yang dihasilkan akan sulit dibaca dalam grafik.
2. Menggunakan windows Xp akan lebih memudahkan kita untuk membaca respon yang dihasilkan oleh sistem pengendali.

DAFTAR PUSTAKA

Gunterus, Frans. *Falsafah Dasar : Sistem Pengendalian Proses*, PT. Alex Media Komputindo, 1994.

Malvino, Albert Paul. *Prinsip-prinsip Elektronika edisi ketiga* – terjemahan, Prof.

M. Barmawi, Erlangga, Jakarta. 1985.

Prasetia Retna, Widodo catur E. *Interfacing Port Paralel dan Port Serial dengan Visual Basic 6.0*, Penerbit ANDI, Jakarta, 2004.

Rusli, Mohammad. *Sistem Kontrol Kedua Automatik* – Terjemahan, Ir. Edi Laksono, Erlangga, Jakarta, 1991.

Seborg Dale E, Edgar Thomas F, Mellichamp Duncanna A, *Process Dynamics And Control* John Willey And Sons, New York 1989.

Killan, *Motor Control Technology : Components and System 2nd*

[www. Google.com](http://www.Google.com)



'program

'=====

'Deklarasi Konstanta Grafik

Const XMin = 150

Const XMax = 20150

Const YMin = 2

Const YMax = 102

Const MaxHitung = 20000

'=====

'=====

'Deklarasi Variabel

Dim PB As Integer 'variabel untuk menentukan besarnya gain

Dim TI As Single 'variabel untuk menentukan besarnya time integral

Dim TD As Single 'variabel untuk menentukan besarnya time differensial

Dim SP, cSP As Integer 'variabel untuk menentukan besarnya set point

Dim MV, cMV As Single 'variabel untuk menampung hasil pengolahan data PID

Dim PV, cPV As Single 'variabel yang menampung hasil sampling dari alat

Dim StartAuto As Boolean 'variabel kontrol program

Dim StartManual As Boolean 'variabel kontrol program

Dim HslProp As Single 'variabel untuk menampung hasil perhitungan
proportional

Dim HslInt As Single 'variabel untuk menampung hasil perhitungan integral

Dim HslTur As Single 'variabel untuk menampung hasil perhitungan turunan

Dim HslKC As Single 'variabel untuk menampung hasil perhitungan KC (gain)

Dim DataAwal As Single 'variabel yang menampung error mula-mula

Dim DataAkhir As Single 'variabel untuk menampung error terbaru

Dim Pilih As Byte 'kontrol untuk memilih struktur PID

Dim HslPID As Single 'variabel sementara untuk menyimpan data perhitungan

Dim Kounter As Integer 'variabel pencacah

Dim cKounter As Integer 'variabel pencacah

Dim Xstep As Integer 'variabel pencacah

Dim T As Single 'Variabel untuk sampling rate

Dim I As Double

Dim A As Double

Dim ExcelApp As Excel.Application 'inisialisasi excell

Dim ExcelWkb As Excel.Workbook 'inisialisasi workbook

Dim ExcelSht As Excel.Worksheet 'inisialisasi worksheet

Dim MyExcel As Boolean 'inisialisasi file excell

'Deklarasi Alamat input dan output

Const OutData = &H378

Const InData = &H379

Const Control = &H37A

Const AlamatData1 = &H4

Const AlamatData2 = &H0

Const AlamatData3 = &H5

Const AlamatData4 = &H1

Const AlamatData5 = &H6

Const AlamatData6 = &H2

Const AlamatMotor = &HB

'Deklarasi Variabel input dan output

Dim Dummy As Integer

Dim PortData As Integer

Dim DH, DL As Byte

Dim Hasil As Single

Dim Data1, Data2, Data3, Data4, Data5, Data6 As Integer

Dim Cacahan_Pulsa, bytePV1, Temp As Double

Private Sub cmdClear_Click()

Dim e As Integer

e = MsgBox("Anda Yakin Akan Membersihkan Layar Grafik", vbOKCancel, "Bersihkan Layar")

If e = vbOK Then

GridLine

cKounter = 0

Kounter = 0

End If

End Sub

Private Sub cmdKeluar_Click()

'=====

'Sub untuk keluar program

'=====

Dim e As Integer

e = MsgBox("Anda akan Keluar dari Program ini ?", vbOKCancel, "Keluar Program")

If e = vbOK Then

MV = 0

KirimData

End

End If

End Sub

Private Sub cmdStart_Click()

```
'=====
'Sub untuk melakuakn proses realtime
'=====
```

Dim BaruT As Single

'Cek Apakah Sampling rate kosong

If txtRate.Text = "" Then

 BaruT = InputBox("Masukan Harga Time Sampling (Dalam detik)", "Time Sampling")

 txtRate.Text = BaruT

End If

'Masukan waktu sampling rate

T = Val(txtRate.Text)

'Cek Mode Manual atau Auto

PilihOption

'Masukan nilai kondisi awal

SettingAwal

'Ubah tombol Start Jadi Stop

If cmdStart.Caption = "Start" Then

 cmdStart.Caption = "Stop"

 Xstep = MaxHitung / Val(Frame.Text)

'Ubah Start menjadi true sebagai pemicu

If optAuto.Value = True Then

 StartAuto = True

 StartManual = False

 txtRate.Enabled = False

 GridLine

 SistemAuto

Else

 StartAuto = False

 StartManual = True

 txtRate.Enabled = False

 GridLine

 SistemManual

```
End If
Else
  cmdStart.Caption = "Start"
  StartAuto = False
  StartManual = False
  txtRate.Enabled = True
  MV = 0
  KirimData
End If
End Sub
```

```
Private Sub Form_Load()
  '=====
  'Kondisi pada saat di eksekusi
  '=====
  Randomize
  On Error Resume Next
  Err.Clear
  Set ExcelApp = CreateObject("Excel.Application")
  ExcelApp.Visible = False

  Start = False
  optAuto.Value = True
  optParalel.Value = True
  txtRate.Text = 0.1
  Frame.Text = 5000

  MV = 0
  KirimData
  SettingAwal
End Sub
```

```
Private Sub Form_Paint()
  '=====
  'Sub untuk menampilkan bingkai
  '=====
  GridLine

```

```
End Sub
```

```
Private Sub Form_Resize()
  '=====
  'Ubah seluruh koordinat pada picGrafik
  '=====

```

```
picGrafik.ScaleLeft = -XMin
picGrafik.ScaleTop = YMax
picGrafik.ScaleWidth = XMax + XMin
picGrafik.ScaleHeight = -(YMax + YMin)
picGrafik.Refresh
```

```
End Sub
```

```
Private Sub PID()
```

```
'=====
=====
```

```
'Prosedur ini untuk menentukan rumus perhitungan dari struktur PID
```

```
'=====
=====
```

```
HslKC = KC(PB)
HslProp = Proportional(DataAkhir, HslKC)
HslInt = Integral(DataAkhir, HslInt)
HslTur = Turunan(DataAkhir, DataAwal)
```

```
'=====
```

```
If Pilih = 1 Then
```

```
    If HslInt > 100 * TI / T Then HslInt = 100 * TI / T
```

```
    If HslInt < -100 * TI / T Then HslInt = -100 * TI / T
```

```
    HslPID = Paralel(TD, TI, HslProp, HslInt, HslTur, T)
```

```
ElseIf Pilih = 2 Then
```

```
    If HslInt > (100 * TI) / (T * HslKC) Then HslInt = (100 * TI) / (T * HslKC)
```

```
    If HslInt < (-100 * TI) / (T * HslKC) Then HslInt = (-100 * TI) / (T * HslKC)
```

```
    HslPID = Seri(TD, TI, HslProp, HslInt, HslTur, T, HslKC, DataAkhir)
```

```
ElseIf Pilih = 3 Then
```

```
    If HslInt > (100 * TI) / (T * HslKC) Then HslInt = (100 * TI) / (T * HslKC)
```

```
    If HslInt < (-100 * TI) / (T * HslKC) Then HslInt = (-100 * TI) / (T * HslKC)
```

```
    HslPID = Mix(TD, TI, HslProp, HslInt, HslTur, T, HslKC)
```

```
End If
```

```
End Sub
```

```
Private Sub StrukturPID()
```

```
'=====
```

```
'Prosedur ini adalah untuk memilih struktur PID
```

```
'=====
```

```
If optParalel.Value Then
    Pilih = 1
End If
If optSeri.Value Then
    Pilih = 2
End If
If optMix.Value Then
    Pilih = 3
End If
```

```
End Sub
```

```
Private Sub optAuto_Click()
```

```
'=====
'Memilih mode Auto
'=====
```

```
optAuto.Value = True
txtPB.Enabled = True
txtTI.Enabled = True
txtTD.Enabled = True
txtSP.Enabled = True
txtMV.Enabled = False
```

```
End Sub
```

```
Private Sub optManual_Click()
```

```
'=====
'Memilih mode Manual
'=====
```

```
optManual.Value = True
txtPB.Enabled = False
txtTI.Enabled = False
txtTD.Enabled = False
txtSP.Enabled = False
txtMV.Enabled = True
```

```
End Sub
```

```
Private Sub optMix_Click()
```

```
'=====
'Memilih struktur Mix
'=====
```

```
optMix.Value = True
```

```

End Sub
Private Sub optParalel_Click()
    '=====
    'Memilih struktur Paralel
    '=====

    optParalel.Value = True

End Sub
Private Sub optSeri_Click()
    '=====
    'Memilih struktur Seri
    '=====

    optSeri.Value = True

End Sub
Private Sub CekData()
    '-----
    'Prosedur ini untuk mengecek apakah seluruh data yang
    'diperlukan sudah terisi kecuali mode manual
    '-----

    Dim BaruPb, BaruSp As String
    Dim BaruTI, BaruTD As String

    If txtSP.Text = "" Then
        BaruSp = InputBox("Masukan Nilai Set Point", "Set Point")
        txtSP.Text = BaruSp
    End If

    If txtPB.Text = "" Then
        BaruPb = InputBox("Masukan Nilai Proportional Band", "Proportional Band")
        txtPB.Text = BaruPb
    End If

    If txtTI.Text = "" Then
        BaruTI = InputBox("Masukan Nilai Time Integral", "Time Integral")
        txtTI.Text = BaruTI
    End If

    If txtTD.Text = "" Then
        BaruTD = InputBox("Masukan Nilai Time Differential", "Time Differential")
        txtTD.Text = BaruTD
    End If

```

```

End If

End Sub

Private Sub PilihOption()
'=====
'Prosedure untuk menentukan apakah kontrol dilakukan secara auto atau manual
'=====

If optAuto.Value Then
    txtPB.Enabled = True
    txtTI.Enabled = True
    txtTD.Enabled = True
    txtSP.Enabled = True
    txtMV.Enabled = False
End If

If optManual.Value Then
    txtPB.Enabled = False
    txtTI.Enabled = False
    txtTD.Enabled = False
    txtSP.Enabled = False
    txtMV.Enabled = True
End If
End Sub

Private Sub picGrafik_DblClick()
'=====
'Prosedur ini untuk merefres/membersihkan tampilan pada layar
'=====

Dim e As Integer

e = MsgBox("Anda Akan Membersihkan Layar Grafik", vbOKCancel, "Bersihkan
Layar")
If e = vbOK Then
    GridLine
    cKounter = 0
    Kounter = 0
End If

```

End Sub

Private Sub SistemAuto()

```
'=====
'Jika pilihoption memilih mode auto
'=====
```

Dim Cadangan, Cadangan2 As Single

AwalData

I = 0

Do While StartAuto = True

 'Apakah seluruh data sudah terisi?

 CekData

 'Tipe struktur apa yang akan dipakai?

 StrukturPID

 'Masukan nilai yang diperlukan untuk PID

 TI = Val(txtTI.Text)

 TD = Val(txtTD.Text)

 SP = Val(txtSP.Text)

 PB = Val(txtPB.Text)

 'Tentukan Error dari PID

 DataAkhir = SP - PV

 'Masukan ke persamaan PID

 PID

 Cadangan = HslPID

 'Batasi nilai MV agar tidak melebihi 100 dan kurang dari 0

 If Cadangan >= 100 Then

 MV = 100

 ElseIf Cadangan < -100 Then

 MV = -100

 Else

 MV = Cadangan

 End If

 'Kirim data ke plant

 KirimData

 'Tampilkan di layar MV

 txtMV.Text = MV


```
'Tampilkan dilayar PV
txtPV.Text = PV
```

```
'Tampilkan juga di grafik
Grafik
```

```
'Simpan seluruh nilai
cKounter = Kounter
cMV = MV
cPV = PV
cSP = SP
```

```
'Naikkan Kounter
Kounter = Kounter + Xstep
```

```
'Cek jangan sampai kounter melebihi angka 10000
If Kounter > MaxHitung Then
    Kounter = 0
    cKounter = 0
    GridLine
End If
```

```
'Ambil data dari plant
TerimaData
PV = Cacahan_Pulsa / 7681536
Tunda (T * 1000)
I = I + T
```

```
    SimpanData
Loop
SaveFile
End Sub
```

```
Private Sub SistemManual()
```

```
    '=====
    'Jika pilihoption memilih mode manual
    '=====
```

```
    AwalData
```

```
    I = 0
```

```
    Do While StartManual = True
```

```
        'Ambil nilai MV pada PID
```

```
        MV = Val(txtMV.Text)
```

```
        'Batasi nilai MV agar tidak melebihi 100
```

```
        If MV > 100 Then
```

```

    MsgBox "Nilai yang Anda Masukan Terlalu Besar (Maksimal 100)",
vbOKOnly, "Salah"
    Start = False
    tmrManual.Enabled = False
    cmdStart.SetFocus
    MV = 0
    KirimData
    Exit Sub
End If

'Kirim nilai MV ke plant
KirimData

'Tampilkan angka di layar PV
txtPV.Text = PV

'Tampilkan juga dalam bentuk grafik
Grafik

'Simpan seluruh nilai
cKounter = Kounter
cMV = MV
cPV = PV
cSP = SP

'Naikkan Kounter
Kounter = Kounter + Xstep

'Cek jangan sampai kounter melebihi angka 10000
If Kounter > MaxHitung Then
    Kounter = 0
    cKounter = 0
    GridLine
End If

'Ambil data dari plant
TerimaData
PV = Cacahan_Pulsa / 768
Tunda (T * 1000)
I = I + T

    SimpanData
Loop
SaveFile
End Sub

```

```

Private Sub txtPB_DblClick()
'=====
'Sub untuk mengganti nilai PB
'=====

Dim BaruPb As String

BaruPb = InputBox("Masukan Nilai Proportional Band Baru", "Nilai Proportional
Band Baru")
txtPB.Text = BaruPb

End Sub

Private Sub txtSP_DblClick()
'=====
'Sub Mengganti nilai SP
'=====

Dim BaruSp As String

BaruSp = InputBox(" Masukan Harga Set Point Baru :", "Nilai Set Point Baru")
txtSP.Text = BaruSp

End Sub

Private Sub SettingAwal()
'=====
'Setting kondisi awal
'=====

Kounter = 0
PV = 0
cPV = 0
cMV = 0
cKounter = 0
SP = 0
cSP = 0
MV = 0
DataAwal = 0
DataAkhir = 0

End Sub

Private Sub Grafik()
'=====
'Prosedur ini untuk menggambar grafik

```

```
'Warna kuning adalah set point
'Warna hijau dan merah adalah MV
'Warna biru adalah PV
```

```
=====
Dim YSP, YcSP As Integer
Dim YMV, YcMV As Single
Dim YPV, YcPV As Single
Dim X, cX As Integer
```

```
'Transfer semua nilai yang diperlukan
X = Kounter 'Ambil nilai X yang baru
cX = cKounter 'Ambil nilai X yang lama
YSP = SP 'Ambil nilai SP yang baru
YcSP = cSP 'Ambil nilai SP yang lama
YMV = MV 'Ambil nilai MV yang baru
YcMV = cMV 'Ambil nilai MV yang lama
YPV = PV 'Ambil nilai PV yang baru
YcPV = cPV 'Ambil nilai PV yang lama
```

```
'Gambar Garis pada grafik
```

```
picGrafik.Line (cX, YcSP)-(X, YSP), vbYellow 'Gambar grafik SP
picGrafik.Line (cX, YcPV)-(X, YPV), vbBlue 'Gambar grafik PV
If MV >= 0 Then
    picGrafik.Line (cX, YcMV)-(X, YMV), vbGreen 'Gambar grafik MV
Else
    YMV = -1 * YMV
    YcMV = -1 * YcMV
    picGrafik.Line (cX, YcMV)-(X, YMV), vbRed 'Gambar grafik MV
End If
End Sub
```

```
Private Sub KirimData()
```

```
=====
'Sub ini dipakai untuk mengirim data ke LPT
=====
```

```
Dim Kirim As Byte
```

```
Kirim = ConvDAC(MV)
Port_Out OutData, Kirim
Port_Out Control, AlamatMotor
```

```
End Sub
```

Private Sub TerimaData()

```
'=====
'Sub untuk menerima data dari LPT
'=====
Port_Out Control, AlamatData1
PortData = Port_In(InData)
Data1 = (PortData And 120) / 8
Port_Out Control, AlamatData2
PortData = Port_In(InData)
Data2 = (PortData And 120) / 8
Port_Out Control, AlamatData3
PortData = Port_In(InData)
Data3 = (PortData And 120) / 8
Port_Out Control, AlamatData4
PortData = Port_In(InData)
Data4 = (PortData And 120) / 8
Port_Out Control, AlamatData5
PortData = Port_In(InData)
Data5 = (PortData And 120) / 8
Port_Out Control, AlamatData6
PortData = Port_In(InData)
Data6 = (PortData And 120) / 8
Cacahan_Pulsa = Data1 + 16 * Data2 + 256 * Data3 + 4096 * Data4 + 65536 *
Data5 + 1048576 * Data6
If Cacahan_Pulsa > 200000 Then Cacahan_Pulsa = 0
End Sub
```

Private Sub GridLine()

```
'=====
'Sub untuk menggambar bingkai grafik dan gridline
'=====
Dim XTab, YTab, X, Y As Integer

picGrafik.Cls
picGrafik.ForeColor = vbBlack

'Gambar bingkai Grafik
picGrafik.Line (0, 0)-(XMax - XMin, YMax - YMin), , B

'Gambar Gridline
XTab = (XMax - XMin) / 10
YTab = (YMax - YMin) / 10

'Gambar gridline vertikal
For X = XTab To (XMax - XMin) - XTab Step XTab
```

```

        picGrafik.Line (X, 0)-(X, YMax - YMin)
    Next

    'Gambar gridline Horizontal
    For Y = YTab To (YMax - YMin) - YTab Step YTab
        picGrafik.Line (0, Y)-(XMax - XMin, Y)
    Next

End Sub

Private Sub SimpanData()
'simpan data Proses KE excell

    'ExcelSht.Cells(a, 1) = Format(I, "##0.0")
    'ExcelSht.Cells(a, 2) = Format(SP, "##0.0")
    'ExcelSht.Cells(a, 3) = Format(MV, "##0.0")
    'ExcelSht.Cells(a, 4) = Format(PV, "##0.0")

    ExcelSht.Cells(A, 1) = I
    ExcelSht.Cells(A, 2) = SP
    ExcelSht.Cells(A, 3) = MV
    ExcelSht.Cells(A, 4) = PV
    A = A + 1
End Sub

Private Sub AwalData()

On Error Resume Next
Err.Clear

    A = 3
    b = 1

Set ExcelWkb = ExcelApp.Workbooks.Add      'aktifkan workbook ecell
Set ExcelSht = ExcelWkb.Worksheets(1)     'aktifkan workseet ecell
MyExcel = True
With ExcelSht                               'pengaturan nama pada kolom ecell
    .Cells.ColumnWidth = 9
    .Cells(1, 1) = "Diambil pada :"
    .Cells(1, 2) = Date
    .Cells(1, 3) = Time()
    .Cells(2, 1) = "I"
    .Cells(2, 2) = "SP"
    .Cells(2, 3) = "MV"
    .Cells(2, 4) = "PV"

```

```
End With  
End Sub
```

```
Private Sub SaveFile()  
Dim c, d, f, G As String  
Dim e As Integer  
e = 1
```

```
If Len(Dir(App.Path & "\Data\data.xls")) <> 0 Then  
On Error Resume Next  
Err.Clear
```

```
G = "\Data\data" & e & ".xls"  
Do While Len(Dir(App.Path & G)) <> 0  
G = "\data\data" & e & ".xls"  
If Len(Dir(App.Path & G)) = 0 Then  
Exit Do  
End If  
e = e + 1  
Loop  
ExcelWkb.SaveAs App.Path & G  
Else  
MkDir (App.Path & "\Data")  
ExcelWkb.SaveAs App.Path & "\Data\data.xls"  
End If  
ExcelWkb.Close True  
If MyExcel Then  
ExcelApp.Quit  
End If  
End Sub
```

Option Explicit

'=====

'Deklarasi file DLL.

Public Declare Sub Tunda Lib "Port_IO.dll" (ByVal lama As Integer)

Public Declare Sub Port_Out Lib "Port_IO.dll" (ByVal nPort As Integer, ByVal
nData As Byte)

Public Declare Function Port_In Lib "Port_IO.dll" (ByVal nPort As Integer) As Byte

Public Declare Function Shift_Kiri Lib "Port_IO.dll" (ByVal Data As Byte, ByVal
kali As Byte) As Byte

Public Declare Function Shift_Kanan Lib "Port_IO.dll" (ByVal Data As Byte, ByVal
kali As Byte) As Byte

Const Ratio = 255

Function Integral(ByVal Data As Single, ByVal Jumlah As Single) As Single

'=====

'Fungsi integral yaitu menjumlah segala masukan

'=====

Integral = Jumlah + Data

End Function

Function Turunan(ByVal Data As Single, ByVal Data_1 As Single) As Single

'=====

'Fungsi Turunan yakni mengurangi data baru

'dengan data lama

'=====

Turunan = Data_1 - Data

End Function

Function Proportional(ByVal Data As Single, ByVal Gain As Single) As Single


```

=====
'Fungsi porportional yakni mengalikan data yang
'masuk dengan sebuah konstanta (Gain)
=====

Proportional = Gain * Data
End Function

Function KC(ByVal Masukan As Integer) As Single
'=====
'Fungsi untuk menghitung besar gain dari PB
'=====

KC = 100 / Masukan
End Function

Function Paralel(ByVal Time_D As Single, ByVal Time_I As Single, ByVal In_Prop
As Single, ByVal In_Integ As Single, ByVal In_Tur As Single, ByVal T As Single)
As Single
'=====
'Fungsi untuk struktur paralel
'=====

Paralel = In_Prop + ((T / Time_I) * In_Integ) + ((Time_D / T) * In_Tur)
End Function

Function Seri(ByVal Time_D As Single, ByVal Time_I As Single, ByVal In_Prop
As Single, ByVal In_Integ As Single, ByVal In_Tur As Single, ByVal T As Single,
ByVal KC As Single, ByVal Data As Single) As Single
'=====
'Fungsi untuk struktur seri
'=====

```

```

    Seri = In_Prop + (((KC * Time_D) / Time_I) * Data) + (((KC * T) / Time_I) *
In_Integ) + (((KC * Time_D) / T) * In_Tur)
End Function

```

```

Function Mix(ByVal Time_D As Single, ByVal Time_I As Single, ByVal In_Prop
As Single, ByVal In_Integ As Single, ByVal In_Tur As Single, ByVal T As Single,
ByVal KC As Single) As Single

```

```

'=====
'Fungsi untuk struktur Mix
'=====
    Mix = In_Prop + (((KC * T) / Time_I) * In_Integ) + (((KC * Time_D) / T) *
In_Tur)
End Function

```

```

Function ConvDAC(ByVal Masukan As Single) As Byte

```

```

'=====
'Fungsi untuk mengganti ke bentuk byte
'=====
    If Masukan > 0 Then
        ConvDAC = (Masukan / 100) * 126
    End If
    If Masukan < 0 Then
        Masukan = -1 * Masukan
        ConvDAC = 128 + (Masukan / 100) * 126
    End If

```

```

End Function

```

```

Function ConvADC(ByVal Masukan As Byte) As Single

```

```

'=====
'Fungsi untuk mengubah ke skala 100
'=====

```

ConvADC = (Masukan / 255) * 100

End Function



Dual 1-of-4 Decoder/ Demultiplexer

High-Performance Silicon-Gate CMOS

The MC54/74HC139A is identical in pinout to the LS139. The device inputs are compatible with standard CMOS outputs; with pullup resistors, they are compatible with LSTTL outputs.

This device consists of two independent 1-of-4 decoders, each of which decodes a two-bit Address to one-of-four active-low outputs. Active-low Selects are provided to facilitate the demultiplexing and cascading functions. The demultiplexing function is accomplished by using the Address inputs to select the desired device output, and utilizing the Select as a data input.

- Output Drive Capability: 10 LSTTL Loads
- Outputs Directly Interface to CMOS, NMOS and TTL
- Operating Voltage Range: 2.0 to 6.0 V
- Low Input Current: 1.0 μ A
- High Noise Immunity Characteristic of CMOS Devices
- In Compliance with the Requirements Defined by JEDEC Standard No. 7A
- Chip Complexity: 100 FETs or 25 Equivalent Gates

MC54/74HC139A



J SUFFIX
CERAMIC PACKAGE
CASE 620-10



N SUFFIX
PLASTIC PACKAGE
CASE 648-08

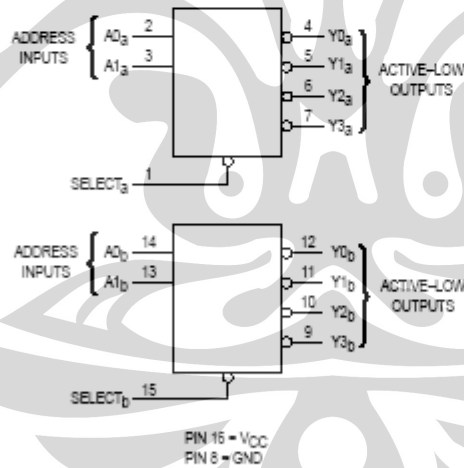


D SUFFIX
SOIC PACKAGE
CASE 751B-05

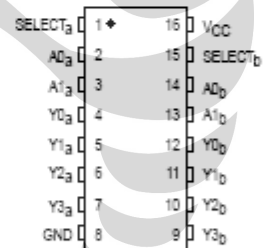
ORDERING INFORMATION

MC54HCXXXAJ Ceramic
MC74HCXXXAN Plastic
MC74HCXXXAD SOIC

LOGIC DIAGRAM



PIN ASSIGNMENT



FUNCTION TABLE

Select	Inputs		Outputs			
	A1	A0	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

X = don't care

Octal 3-State Noninverting Buffer/Line Driver/ Line Receiver

High-Performance Silicon-Gate CMOS

The MC54/74HC244A is identical in pinout to the LS244. The device inputs are compatible with standard CMOS outputs; with pullup resistors, they are compatible with LSTTL outputs.

This octal noninverting buffer/line driver/line receiver is designed to be used with 3-state memory address drivers, clock drivers, and other bus-oriented systems. The device has noninverting outputs and two active-low output enables.

The HC244A is similar in function to the HC240A and HC241A.

- Output Drive Capability: 15 LSTTL Loads
- Outputs Directly Interface to CMOS, NMOS, and TTL
- Operating Voltage Range: 2 to 6 V
- Low Input Current: 1 μ A
- High Noise Immunity Characteristic of CMOS Devices
- In Compliance with the Requirements Defined by JEDEC Standard No. 7A
- Chip Complexity: 138 FETs or 34 Equivalent Gates

MC54/74HC244A



J SUFFIX
CERAMIC PACKAGE
CASE 732-03



N SUFFIX
PLASTIC PACKAGE
CASE 738-03



DW SUFFIX
SOIC PACKAGE
CASE 751D-04



SD SUFFIX
SSOP PACKAGE
CASE 940C-03

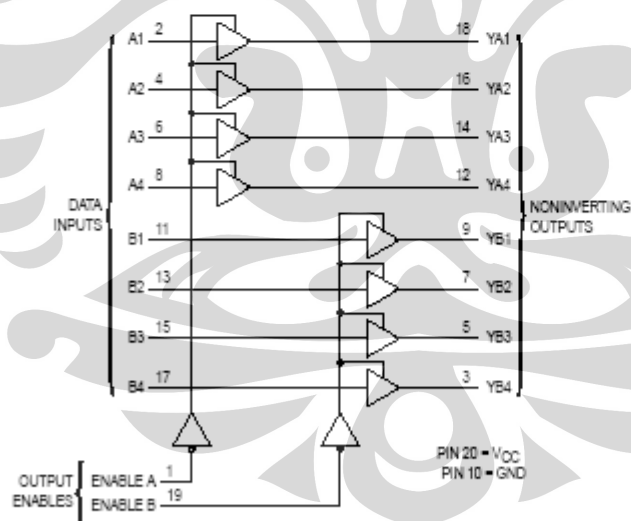


DT SUFFIX
TSSOP PACKAGE
CASE 943E-02

ORDERING INFORMATION

MC54HCXXXAJ	Ceramic
MC74HCXXXAN	Plastic
MC74HCXXXADW	SOIC
MC74HCXXXASD	SSOP
MC74HCXXXADT	TSSOP

LOGIC DIAGRAM



PIN ASSIGNMENT

ENABLE A	1	20	VCC
A1	2	19	ENABLE B
YB4	3	18	YA1
A2	4	17	B4
YB3	5	16	YA2
A3	6	15	B3
YB2	7	14	YA3
A4	8	13	B2
YB1	9	12	YA4
GND	10	11	B1

FUNCTION TABLE

Inputs		Outputs
Enable A, Enable B	A, B	YA, YB
L	L	L
L	H	H
H	X	Z

Z = high impedance



Octal 3-State Noninverting Bus Transceiver

High-Performance Silicon-Gate CMOS

The MC54/74HC245A is identical in pinout to the LS245. The device inputs are compatible with standard CMOS outputs; with pullup resistors, they are compatible with LSTTL outputs.

The HC245A is a 3-state noninverting transceiver that is used for 2-way asynchronous communication between data buses. The device has an active-low Output Enable pin, which is used to place the I/O ports into high-impedance states. The Direction control determines whether data flows from A to B or from B to A.

- Output Drive Capability: 15 LSTTL Loads
- Outputs Directly Interface to CMOS, NMOS, and TTL
- Operating Voltage Range: 2 to 6 V
- Low Input Current: 1 μ A
- High Noise Immunity Characteristic of CMOS Devices
- In Compliance with the Requirements Defined by JEDEC Standard No. 7A
- Chip Complexity: 308 FETs or 77 Equivalent Gates

MC54/74HC245A



J SUFFIX
CERAMIC PACKAGE
CASE 732-03



N SUFFIX
PLASTIC PACKAGE
CASE 738-03

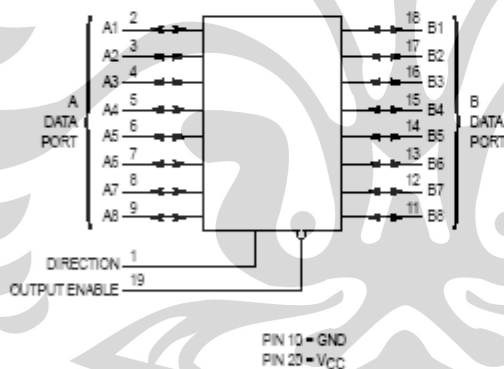


DW SUFFIX
SOIC PACKAGE
CASE 751D-04

ORDERING INFORMATION

MC54HCXXXAJ Ceramic
MC74HCXXXAN Plastic
MC74HCXXXADW SOIC

LOGIC DIAGRAM



PIN ASSIGNMENT

DIRECTION	1	20	Vcc
A1	2	19	OUTPUT ENABLE
A2	3	18	B1
A3	4	17	B2
A4	5	16	B3
A5	6	15	B4
A6	7	14	B5
A7	8	13	B6
A8	9	12	B7
GND	10	11	B8

FUNCTION TABLE

Control Inputs		Operation
Output Enable	Direction	
L	L	Data Transmitted from Bus B to Bus A
L	H	Data Transmitted from Bus A to Bus B
H	X	Buses Isolated (High-Impedance State)

X = don't care

