

BAB II

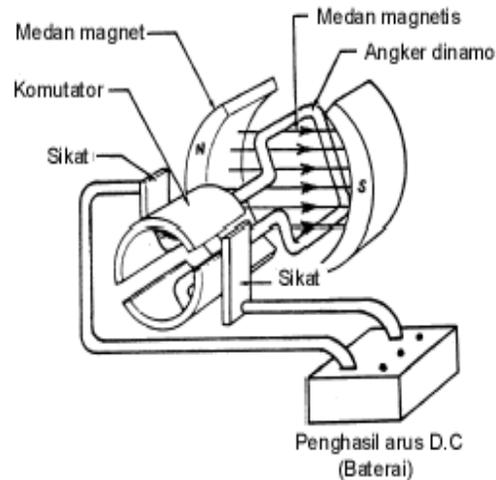
LANDASAN TEORI

Dalam bab ini diuraikan teori yang meliputi *hardware* dan *software* pembentuk alat yang dirancang. *Hardware* yang dimaksud adalah Motor DC, PPI8255, DAC0808, ADC0804, *Tachogenerator*, Rangkaian Penyearah, Rangkaian Pelemah serta *Driver* Motor DC. *Software* yang akan dipakai dan dijelaskan disini adalah Borland Turbo C++ V3.0 dan MATLAB. Selain itu juga diuraikan teori - teori yang berkenaan dengan identifikasi sistem yang dirancang dan dibuat. Selanjutnya akan dijelaskan secara rinci dibawah ini.

2.1 Motor DC

Motor DC (*Direct Current*) adalah sebuah mesin yang mengubah energi listrik menjadi energi mekanis, dengan menggunakan sumber tegangan arus searah atau DC. Motor bekerja karena ada gaya yang bekerja pada konduktor yang dilewatkan medan arus dan medan magnet yang ditimbulkan oleh angker dinamo (Bahan Pelatihan Nasional, 2002). Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

Bentuk Motor DC paling sederhana memiliki kumparan satu lilitan kawat yang bisa berputar bebas di antara kutub – kutub magnet permanen seperti Gambar 2.1 dibawah. Catu tegangan DC dari baterai menuju ke lilitan dengan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan kawat pada gambar di bawah disebut angker dinamo



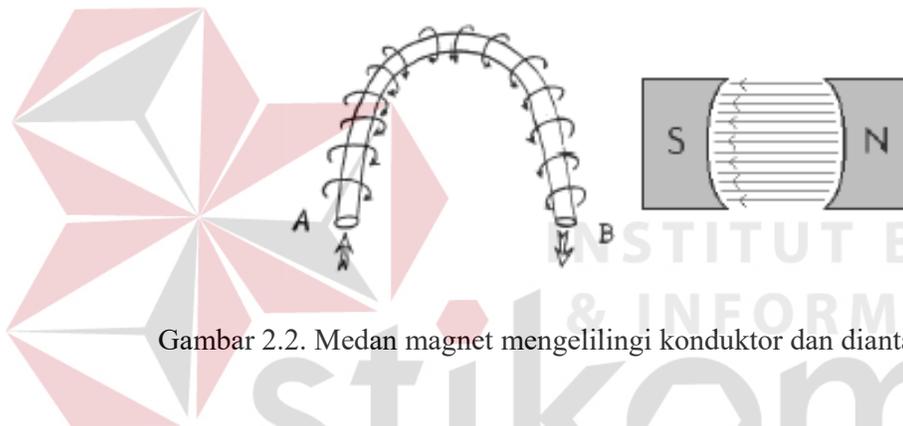
Gambar 2.1 Motor DC sederhana

Komponen – komponen Motor DC sederhana antara lain adalah :

1. Angker dinamo terdiri dari beberapa lilitan kawat yang menghasilkan medan magnet jika dialiri arus. Angker dinamo dapat berputar dengan bebas pada porosnya yang disangga oleh bantalan poros pada kedua ujungnya.
2. Komutator terpasang pada ujung batang atau poros angker dinamo. Komutator terbuat dari dua semen tembaga yang dihubungkan dengan konduktor angker dinamo. Kedua segmen komutator diisolasi satu sama lain maupun terhadap poros. Komutator merubah arah aliran arus menuju angker agar angker dinamo dapat selalu berputar.
3. Sikat biasanya terbuat dari campuran tembaga karbon dan ditempelkan pada komutator dengan didorong oleh pegas. Sikat mengalirkan arus dari catu daya DC pada angker dinamo yang berputar.
4. Kumparan magnet menghasilkan medan magnet di antara kedua kutubnya.

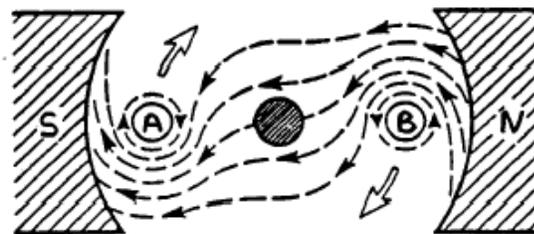
2.1.1 Prinsip kerja Motor DC

Jika arus lewat pada suatu konduktor, timbul medan magnet di sekitar konduktor. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor. Aturan Genggaman Tangan Kanan bisa dipakai untuk menentukan arah garis fluks di sekitar konduktor. Genggam konduktor dengan tangan kanan dengan jempol mengarah pada arah aliran arus, maka jari-jari anda akan menunjukkan arah garis fluks (Bahan Pelatihan Nasional, 2002).



Gambar 2.2. Medan magnet mengelilingi konduktor dan diantara kutub.

Medan magnet hanya terjadi di sekitar sebuah konduktor jika ada arus mengalir pada konduktor tersebut. Jika konduktor berbentuk U (angker dinamo) diletakkan di antara kutub Utara dan Selatan yang kuat medan magnet konduktor akan berinteraksi dengan medan magnet kutub. Lihat gambar di bawah ini :



Gambar 2.3. Reaksi garis fluks.

Lingkaran bertanda A dan B merupakan ujung konduktor yang dilengkungkan (*looped conductor*). Arus mengalir masuk melalui ujung A dan keluar melalui ujung B. Medan konduktor A yang searah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di bawah konduktor. Konduktor akan berusaha bergerak ke atas untuk keluar dari medan kuat ini. Medan konduktor B yang berlawanan arah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di atas konduktor. Konduktor akan berusaha untuk bergerak turun agar keluar dari medan yang kuat tersebut. Gaya - gaya tersebut akan membuat angker dinamo berputar searah jarum jam.

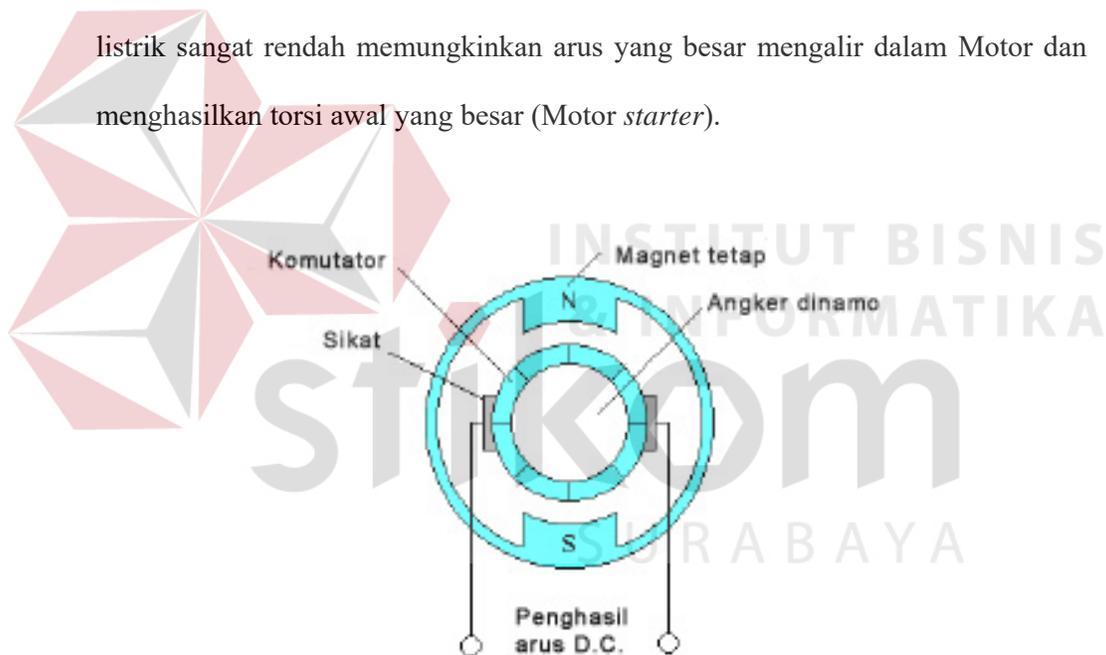
Tegangan catu daya DC dihubungkan pada angker dinamo melalui sikat yang menempel pada komutator. Secara otomatis komutator membalik hubungan antara angker dinamo dan catu daya DC sehingga konduktor berputar pada arah yang tepat terhadap medan magnet, maka angker dinamo akan selalu berputar.

Komutator berputar bersamaan dengan angker dinamo. Jika angker dinamo berputar setengah putaran tiap segmen komutator terputus kontakannya dengan satu sikat dan berganti terhubung dengan sikat yang lain. Oleh karena itu bagian angker dinamo yang paling dekat dengan kutub Selatan akan selalu terdesak ke atas sedangkan yang paling dekat dengan kutub Utara selalu terdorong ke arah bawah. Aksi ini membuat angker dinamo tetap berputar. Jika jumlah lilitan konduktor dan segmen komutator makin banyak maka akan dihasilkan putaran yang lebih kuat.

2.1.2 Kecepatan Motor DC

Kecepatan Motor dipengaruhi oleh kekuatan fluks medan magnet, arus angker dinamo dan jumlah konduktor, serta nilai EMF balik yang ditimbulkan. (Bahan Pelatihan Nasional, 2002). Gambar 2.4 menunjukkan magnet permanen Motor DC. Pengontrolan torsi atau kecepatan dilakukan dengan merubah tegangan catu daya angker dinamo.

Torsi tergantung pada jumlah dan ukuran konduktor pada kedua kumparan maupun pada angker dinamo. Konduktor berat dan besar memiliki resistansi listrik sangat rendah memungkinkan arus yang besar mengalir dalam Motor dan menghasilkan torsi awal yang besar (*Motor starter*).



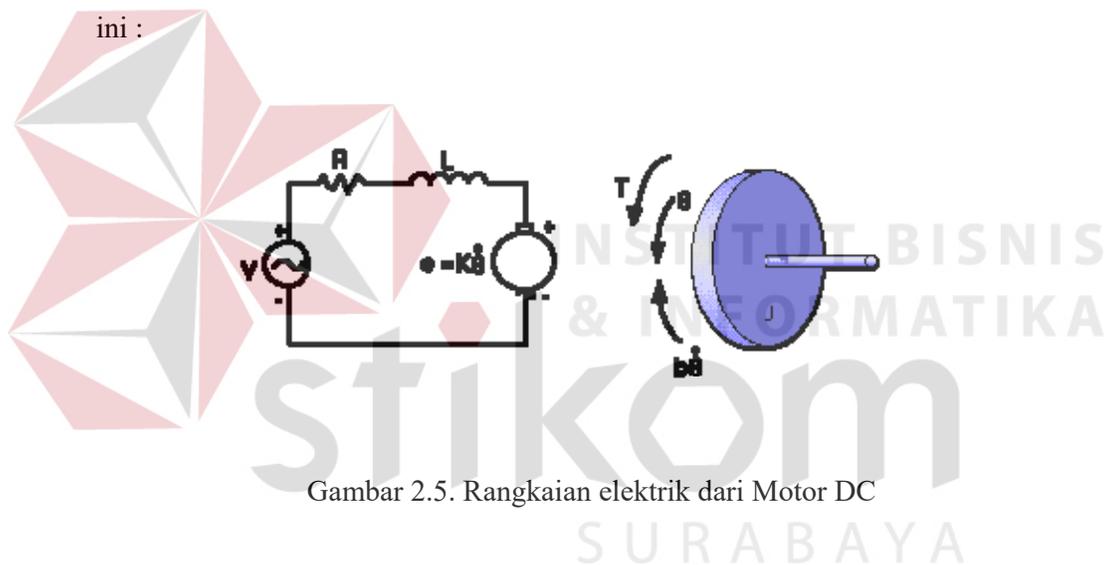
Gambar 2.4. Motor magnet permanen.

Arah putaran angker dinamo tergantung pada posisi kutub medan magnet serta arah arus yang mengalir melewati angker dinamo. Pada Motor DC dengan kumparan magnet permanen, perubahan polaritas pada angker dinamo (arah aliran arus) akan mengubah arah putaran angker dinamo. Kecepatan dan torsi Motor

juga bergantung pada susunan lilitan. Macam lilitan antara lain seri, melangsir dan komposit (gabungan), tiap lilitan mempunyai karakteristik berbeda.

2.1.3 Karakteristik Motor DC

Actuator yang umum di gunakan dalam sistem kontrol adalah Motor DC, karena secara langsung menyediakan gerak putar dan juga dapat menghasilkan gerak transisi (Control Tutorial For MATLAB, 1997), selain itu putaran yang dihasilkan juga halus. Rangkaian elektriknya dapat di gambarkan seperti di bawah ini :



Gambar 2.5. Rangkaian elektrik dari Motor DC

Untuk contoh, dapat diasumsikan nilai-nilai berikut untuk parameter fisik, nilai – nilai ini diperoleh dari percobaan yang diadakan di Carnegie Mellon's Undergraduate Controls Lab (Regents of the University of Michiganian , 1996).

1. *Momen inertia* dari Motor (J) = 0,01 Kg.m²/detik²
2. *Damping ratio* dari sistem mekanik = 0,1 Nm.detik
3. Konstanta elektromotoris ($K = K_e = K_t$) = 0,01 Nm/Amp
4. Tahanan atau *resistor* (R) = 1 Ohm

5. Induktansi (L) = 0,5 H
6. *Input* (V) = sumber tegangan DC (Volt)
7. *Output* (Θ) = kecepatan putaran motor (rpm)

Torsi Motor T , di hubungkan dengan arus jangkar i , konstanta K_t , EMF balik e , di hubungkan dengan percepatan putaran, akan menghasilkan suatu persamaan :

$$T = K_t i \quad (2.1)$$

$$e = K_e \dot{\theta} \quad (2.2)$$

Dalam satuan SI, K_t (konstanta *armature*) sama dengan K_e (konstanta Motor), dari gambar 2.5 didapat persamaan yang berdasarkan pada Hukum ke -3 Newton, yaitu setiap gaya yang diadakan pada suatu benda, menimbulkan gaya lain yang sama besarnya dengan gaya tadi, namun berlawanan arah (Darmawan, 1988 : 34) dan Hukum Kirchhoff untuk rangkaian, yaitu dalam lintasan (*loop*) yang tertutup, jumlah aljabar beda potensial adalah nol. Dengan catatan bila potensial naik, beda potensial dihitung positif dan bila potensial turun dihitung negatif (Darmawan, 1988 : 208) sebagai berikut :

$$J \ddot{\theta} + b \dot{\theta} = K i \quad (2.3)$$

$$L \frac{di}{dt} + R i = V - K \dot{\theta} \quad (2.4)$$

Dari rumus diatas, dengan transformasi Laplace dalam domain s dapat ditulis sebagai berikut :

$$s(Js + b)\Theta(s) = KI(s) \quad (2.5)$$

$$(Ls + R)I(s) = V - Ks\Theta(s) \quad (2.6)$$

Karena dua persamaan tersebut sama – sama mempunyai nilai $I(s)$, maka kedua persamaan tersebut dapat di gabung dan menghasilkan fungsi alih dari *open loop system*, dimana putaran baling - baling Motor sebagai *output* dan tegangan sebagai *input* :

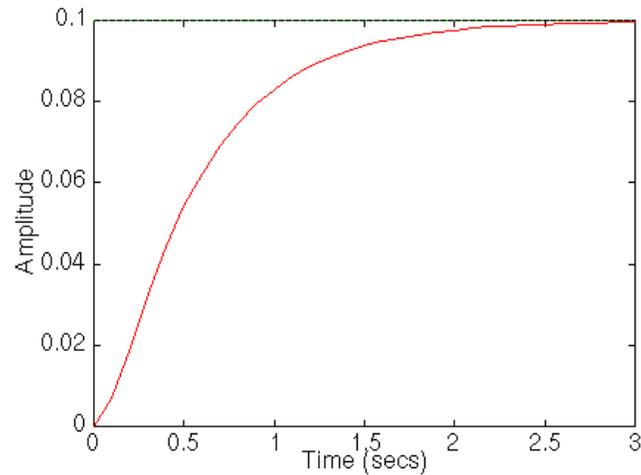
$$Ks\Theta(s) + (Ls + R)\left(\frac{s(Js + b)\Theta(s)}{K}\right) = V \quad (2.7)$$

$$\frac{\Theta}{V} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad (2.8)$$

Dengan MATLAB sesuai fungsi alih di atas dapat di peroleh gambar grafik *step response open loop system* seperti Gambar 2.6 dibawah. Dari persamaan fungsi alihnya dapat diperoleh persamaan *state space*-nya dengan menggunakan kecepatan putaran dan arus listrik sebagai *state variable* dan tegangan sebagai *input*, *output*-nya nanti dalam bentuk kecepatan putaran :

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{j} & \frac{K}{j} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V \quad (2.9)$$

$$\dot{\theta} = [1 \quad 0] \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} \quad (2.10)$$



Gambar 2.6. *Step response* untuk *open loop system*

Grafik diatas merupakan perwujudan dari fungsi alih yang didapat dari persamaan Motor DC umum.

2.2 Fungsi Alih

Dalam teori kontrol, fungsi yang disebut “Fungsi Alih” seringkali digunakan untuk mencirikan hubungan masukan – keluaran dari sistem linier parameter konstan. Konsep fungsi alih hanya digunakan pada sistem linier parameter konstan, walaupun dapat diperluas untuk suatu sistem kontrol nonlinier. “Fungsi alih didefinisikan secara jelas sebagai perbandingan dari Transformasi Laplace keluaran (fungsi tanggapan) dan Transformasi Laplace masukan (fungsi penggerak)” (Astrom dan Wittenmark, 1997), dengan anggapan bahwa semua syarat awal adalah nol.

$$a_0^{(n)} y + a_1^{(n-1)} \dot{y} + \dots + a_{n-1} \ddot{y} + a_n y = b_0^{(m)} x + b_1^{(m-1)} \dot{x} + \dots + b_{m-1} \ddot{x} + b_m x \quad (n \geq m) \quad (2.11)$$

Tinjau sistem linier parameter konstan yang didefinisikan persamaan diferensial (2.12), dimana y adalah keluaran sistem dan x adalah masukannya.

Fungsi alih dari sistem ini diperoleh dengan mencari Transformasi Laplace dari kedua ruas persamaan, dengan menganggap bahwa semua syarat adalah nol

$$\text{Fungsi alih} = G(s) = \frac{Y(s)}{X(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (2.12)$$

Fungsi alih adalah suatu ekspresi yang merelasikan keluaran dan masukan suatu sistem linier parameter konstan dalam bentuk parameter sistem dan merupakan sifat dari sistem itu sendiri, tidak bergantung pada fungsi masukan atau penggerak. Fungsi alih mencakup satuan – satuan yang diperlukan untuk merelasikan masukan dengan keluaran.

Dengan menggunakan konsep ini, kita dapat menyatakan dinamika sistem dengan beberapa persamaan aljabar dalam *domain* s . Pangkat tertinggi dari s pada penyebut fungsi alih sama dengan *orde* suku turunan tertinggi dari keluaran. Jika pangkat tertinggi dari s tersebut adalah n , maka sistem tersebut disebut sistem *orde* ke n .

Hal lain yang berhubungan dengan fungsi alih adalah jenis masukan yang akan diberikan kepada suatu sistem yang menjadi objek. Ada 2 cara yang dapat digunakan, dengan memberikan masukan berupa *Unit Step* dan masukan PRBS (*Pseudo Random Binary Sequence*).

2.2.1 *Step input (tanggapan tangga atau unit step)*

Suatu aspek keluaran $y(t)$ yang disebabkan oleh masukan $x(t)$ disebut sebagai tanggapan dari sistem. Meskipun banyak sekali jenis masukan yang dapat diberikan kepada sistem, tetapi untuk keperluan evaluasi dan analisa sistem, masukan berikut adalah sangat penting, yakni :

1. Masukan tangga (*step input*)

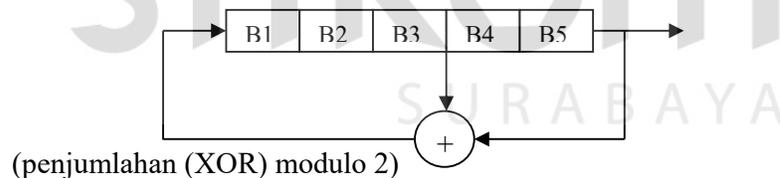
2. Masukan *impulse* (*impulse input*)
3. Masukan *sinusoidal* (*sinusoidal input*)

Point 1 dan 2 sering dipakai untuk menyelidiki tanggapan peralihan dan karenanya menjadi sangat penting. Tanggapan sistem merupakan alat bantu untuk mendefinisikan sifat dari sistem, sebagai pertimbangan penentu apakah sifat dari sistem tersebut diinginkan atau tidak diinginkan dalam konteks tertentu yang diketahui.

2.2.2 PRBS

PRBS dibangkitkan oleh *shift register* dengan *feedback* yang diimplementasikan pada *hardware* dan *software*. Panjang maksimum deret adalah $2^N - 1$ dimana N adalah jumlah sel pada *shift register*.

Gambar 2.7 dibawah menggambarkan pembangkitan PRBS dengan panjang deret $31 = 2^5 - 1$, didapat dengan menggunakan sebuah *shift register* 5 bit.



Gambar 2.7. Pembangkitan PRBS panjang $2^5 - 1$

Perhatikan bahwa sekurang - kurangnya satu sel dari N sel *shift register* seharusnya mempunyai nilai logika tidak sama dengan nol (satu secara umum membuat semua nilai awal dari N sel sama dengan nilai logika 1/XOR).

Perhatikan bahwa satu karakteristik elemen yang sangat penting dari PRBS adalah durasi maksimum dari *impuls* PRBS adalah sama dengan N (jumlah

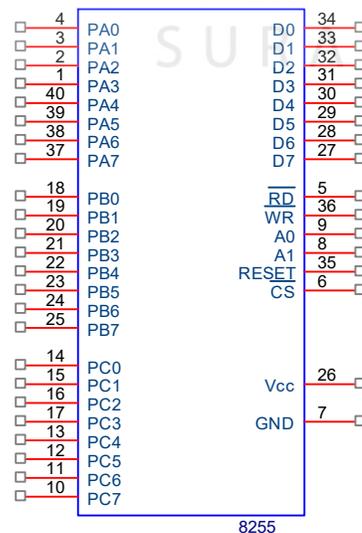
sel). Sifat ini harus dipertimbangkan ketika memilih sebuah PRBS untuk identifikasi sistem.

2.3 PPI 8255

PPI (*Programmable Peripheral Interface*) 8255 merupakan salah satu *interface* yang umum digunakan dalam sistem kontrol, sebagai penghubung komputer dengan *peripheral* di luarnya. Keunggulan PPI 8255 antara lain adalah :

1. 24 Bit *Programmable I/O*
2. Kompatibel dengan semua TTL (*Transistor - Transistor Logic*)
3. Kompatibel dengan semua prosesor INTEL
4. Tersedia dalam 40 pin DIP

PPI 8255 didesain untuk peralatan I/O yang dapat diprogram dengan menggunakan mikroprosesor INTEL. IC (*Integrated Circuit*) ini mempunyai 24 pin I/O yang dapat diprogram secara terpisah dalam 2 kelompok yang masing-masing terdiri dari 12 bit dan digunakan dalam 3 mode operasi yang utama.



Gambar 2.8. Konfigurasi Pin PPI 8255

Diskripsi fungsional PPI 8255 dari INTEL adalah *chip parallel* yang sederhana. Keuntungan utamanya adalah *interface* yang sederhana dari 8 bit *bidirectional*. PPI mengandung *Register Control* dan tiga *port* terpisah yang dapat dialamati, disebut *Port A*, *Port B* dan *Port C*.

Diakses tidaknya PPI ini ditentukan dari sinyal pada pin CS (*chip select*) dan arah akses sesuai dengan sinyal RD dan WR. Pin A0 dan A1 menentukan register mana yang dialamati.

Fungsi dan kegunaan dari masing - masing pin tersebut adalah sebagai berikut :

1. Pin 27 sampai 34 sebagai D0 sampai D7 adalah bus data yang dihubungkan dengan D0 sampai D7 pada slot ISA.
2. Pin 35 sebagai *Reset* adalah pin kontrol untuk inialisasi PPI 8255, dihubungkan dengan jalur *Reset* pada slot ISA.
3. Pin 5 sebagai *Read* adalah pin kontrol untuk menandakan operasi baca oleh komputer ke PPI 8255.
4. Pin 36 sebagai *Write* adalah pin kontrol untuk menandakan operasi tulis dari komputer ke PPI 8255.
5. Pin 6 sebagai *Chips Select* adalah pin kontrol untuk menandakan bahwa PPI 8255 sedang ditunjuk untuk menerima atau mengirim data.
6. Pin 8 sampai 9 sebagai *Port Select* adalah pin alamat untuk memilih *register* yang akan digunakan pada saat pemrograman IC PPI 8255, dihubungkan pada sebagian jalur alamat dari slot ISA.
7. Pin 1 sampai 4 sebagai PA3 sampai PA0 adalah 4 bit jalur data dari 8 bit jalur *Port A* sebagai I/O yang akan dihubungkan ke peralatan luar.

8. Pin 37 sampai 40 sebagai PA7 sampai PA4 adalah 4 bit jalur data dari 8 bit jalur *Port A* sebagai I/O yang akan dihubungkan ke peralatan luar.
9. Pin 18 sampai 25 sebagai PB0 sampai PB7 adalah 8 bit jalur data I/O *Port B*.
10. Pin 10 sampai 17 sebagai PC0 sampai PC7 adalah 8 bit jalur data I/O *Port C*.
11. Pin 7 berfungsi sebagai GND (*Ground*)
12. Pin 26 berfungsi sebagai Vcc (+5V).

Dari Keterangan diatas, maka sebuah IC PPI8255 mempunyai 3 X 8 bit jalur I/O yang dapat digunakan sebagai *input*, *output*, atau kontrol (misalnya untuk sinyal *Strobe*, *Interrupt*) ke atau dari peralatan luar

Komponen IC ini termasuk dalam golongan IC LSI (*Large Scale Integrated*) atau IC yang mempunyai ukuran besar. IC ini dikemas dalam bentuk DIL (*dual in line*) dengan memiliki jumlah pin sebanyak 40 buah dan mempunyai jalur *input output* sebanyak 24 jalur, yang dirancang untuk fungsi I/O pada pemakaian mikrokomputer. Dimana masing-masing pin telah mempunyai fungsi sendiri-sendiri.

Dasar-dasar operasi yang ada pada PPI 8255 berguna untuk menentukan bentuk operasi dari PPI 8255 yang ada pada masing-masing *port*, baik *Port A*, *Port B*, *Port C* ataupun *Control Word* yang dimiliki PPI 8255.

Dasar operasi yang disediakan oleh PPI 8255 merupakan bagian dari prinsip kerja dari masing-masing *port*, yang nantinya menentukan apakah *port* yang digunakan nantinya berfungsi sebagai *input* ataupun *output*, dan untuk *port* terakhir untuk menentukan *Control Word* yang digunakan oleh PPI 8255.

Tabel 2.1 Operasi dasar dari PPI 8255

A1	A2	RD	WR	CS	Operasi Input (READ)
0	0	0	1	0	READ : <i>Port A</i> ke data bus
0	1	0	1	0	READ : <i>Port B</i> ke data bus
1	0	0	1	0	READ : <i>Port C</i> ke data bus
A1	A0	RD	WR	CS	Operasi output (WRITE)
0	0	1	0	0	WRITE: data bus ke <i>Port A</i>
0	1	1	0	0	WRITE: data bus ke <i>Port B</i>
1	0	1	0	0	WRITE: data bus ke <i>Port C</i>
1	1	1	0	0	WRITE : data bus ke <i>Control Word Register</i> (INISIALISASI)
X	X	X	X	1	Data bus 8255 berada dalam kondisi tristate atau PPI tidak difungsikan.
1	1	0	1	0	Kondisi yang tidak mungkin karena CW hanya bisa untuk WRITE.
X	X	1	1	0	Data bus 8255 berada dalam kondisi tristate atau PPI tidak difungsikan.

Mode kerja dari PPI 8255 dioperasikan dalam tiga mode, yaitu:

1. Mode 0 : *Basic Input/Output*

Mode ini disebut juga mode *simple I/O operation*. Pada mode 0 *port* berfungsi sebagai *input* atau *output* tanpa *strobe*. Bila *Port A* dan *B* bekerja dalam mode 0, maka *Port C* dapat bekerja sebagai *port* 8 bit dan juga dapat bekerja sebagai *port* 4 bit secara terpisah. Bila digunakan sebagai *output*, *Port C* secara terpisah dapat di *Set* dan *Reset* dengan mengirim sinyal *Control Word* tertentu ke alamat *Register Control*.

2. Mode 1 : *Strobed Input/Output*

Jika diinginkan *Port A* dan *Port B* bekerja sebagai *input* dan *output strobe*, maka *port* harus di inisialisasi pada mode 1. Pada mode ini beberapa pin dari *Port C* difungsikan sebagai jalur *strobe*. PC0, PC1, PC2 difungsikan sebagai *strobe* untuk *Port B*. Jika *Port A* diinisialisasi sebagai *port output handshake*, maka PC3,

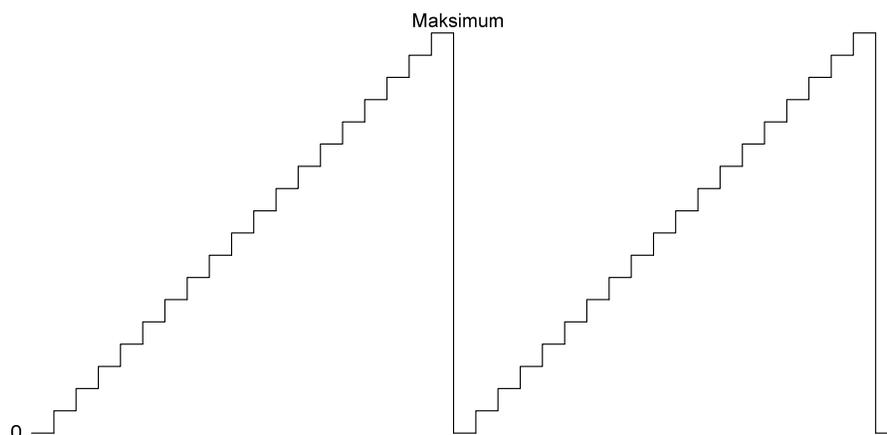
PC6, PC7 berfungsi sebagai sinyal *handshake*. PC4, PC5 dapat digunakan sebagai jalur *input* atau *output*.

3. Mode 2 : *Bidirectional Bus*

Hanya *Port A* yang dapat diinisialisasi pada mode 2. *Port A* dapat digunakan sebagai pemindahan data 2 arah. PC3 sampai PC7 digunakan sebagai jalur *strobe* untuk *Port A* dan PC0 - PC2 dapat digunakan sebagai *input* atau *output* bila *Port B* digunakan dalam mode 0. Bila *Port B* dalam mode 1, maka PC0 - PC2 digunakan sebagai *Strobe* untuk *Port B*.

2.4 DAC 0808

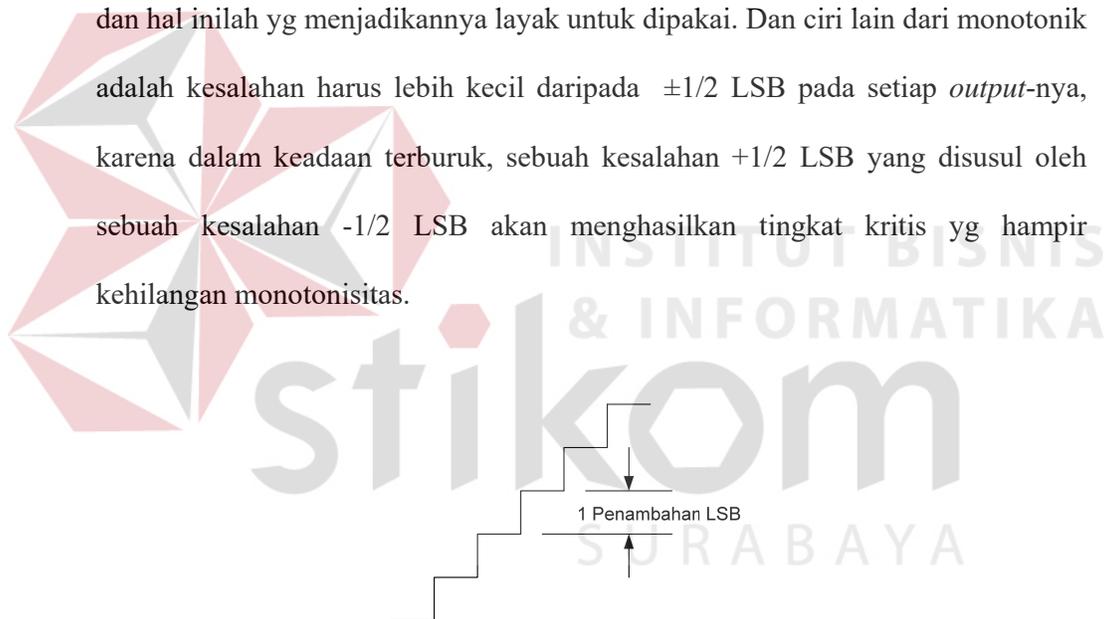
DAC (*Digital To Analog Converter*) berfungsi sebagai media untuk mengkonversikan sinyal digital ke sinyal analog. Sinyal digital disini bisa dari PPI ataupun Mikrokontroler, sedang sinyal analog nantinya akan diproses lebih lanjut oleh Motor ataupun *plant* lainnya. DAC 0808 merupakan jenis DAC yang sering digunakan oleh umum dikarenakan dari segi harga tidak terlalu mahal, dan juga memiliki tingkat ketelitian yang bagus.



Gambar 2.9 Sifat monotonik sebuah DAC.

DAC ini bersifat monotonik dan juga merupakan DAC tangga R-2R 8-bit yang dilengkapi dengan sumber arus acuan dan delapan buah Transistor saklar untuk mengarahkan arus biner. Suatu tegangan dan hambatan eksternal dipergunakan untuk mengatur arus acuan pada nilai yang biasanya bernilai 2mA. DAC ini memiliki waktu pemantapan (*time settling*) 150ns dan ketelitian relatif $\pm 1/2$ LSB.

DAC 0808 bersifat monotonik artinya DAC ini menghasilkan suatu *output* arus yang meningkat untuk setiap masukan digital yang berturut – turut, dan hal inilah yg menjadikannya layak untuk dipakai. Dan ciri lain dari monotonik adalah kesalahan harus lebih kecil daripada $\pm 1/2$ LSB pada setiap *output*-nya, karena dalam keadaan terburuk, sebuah kesalahan $+1/2$ LSB yang disusul oleh sebuah kesalahan $-1/2$ LSB akan menghasilkan tingkat kritis yg hampir kehilangan monotonisitas.

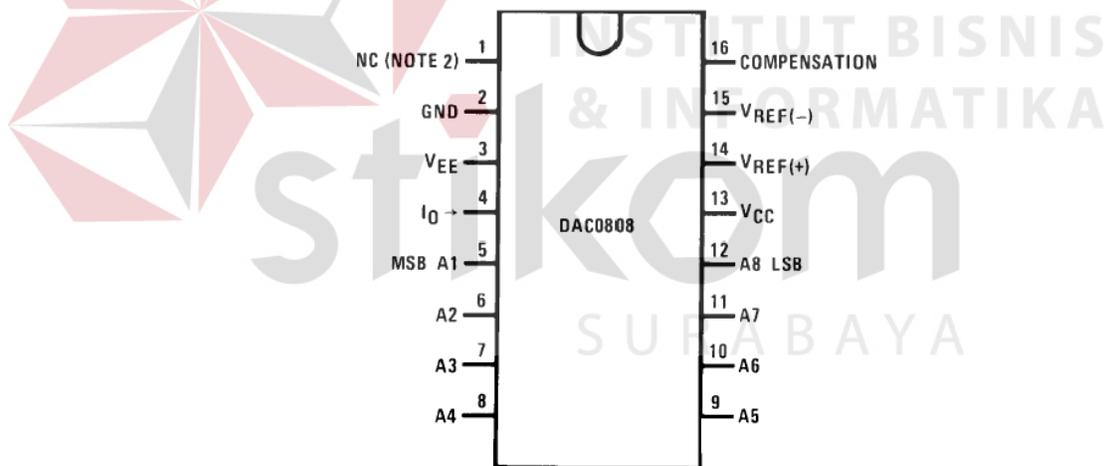


Gambar 2.10 Setiap anak tangga sama dengan penambahan satu LSB

DAC memiliki 16 pin atau kaki, sebagai berikut :

1. NC : *No Connect*, Pin ini tidak memiliki fungsi. Jadi tidak dihubungkan.
2. GND : *Ground*, sebagai kutub negatif.
3. V_{EE} : Dihubungkan dengan catu daya -15V.

4. I_{OUT} : Saluran balik dari tanah (*ground return*). Pin ini dihubungkan dengan Op-Amp.
5. D7-D0 : *Input* data dari sinyal digital 8 bit.
6. V_{CC} : Dihubungkan dengan catu daya +5V.
7. $V_{REF(+)}$: dihubungkan dengan catu daya positif melalui R yang memungkinkan anda untuk menetapkan arus acuan (I_{REF}) pada harga 2mA.
8. $V_{REF(-)}$: dihubungkan ke *ground* melalui R.
9. Kompensasi frekwensi : sebuah Kapasitor yang di hubungkan dengan Pin 16 dan Pin 3 berfungsi untuk memberikan frekuensi yang cocok untuk DAC ini.



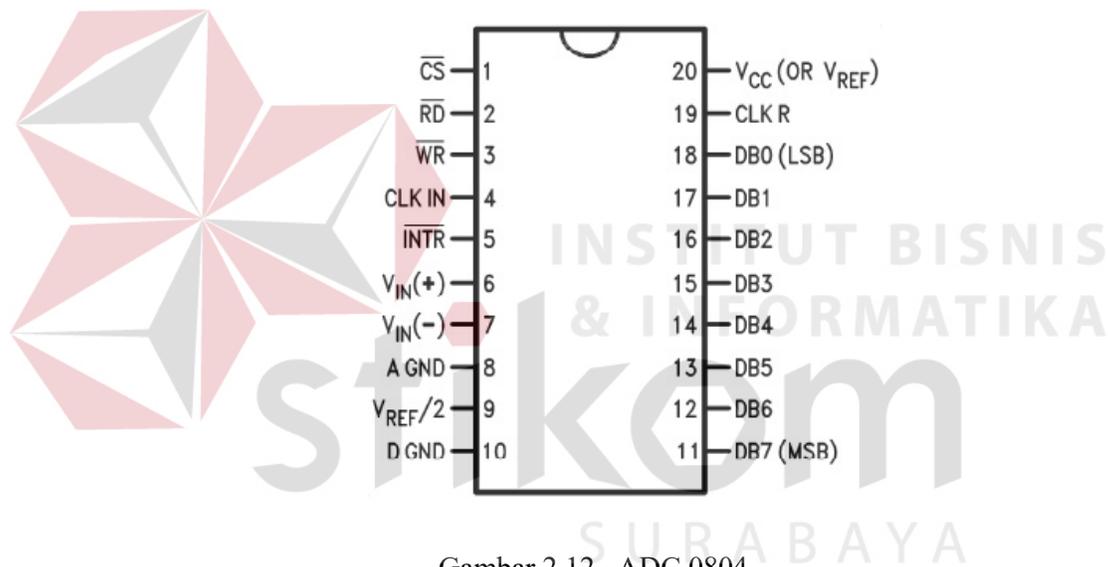
Gambar 2.11. DAC 0808

Waktu pemantapan diartikan sebagai waktu yg dibutuhkan untuk stabilisasi keluaran konverter dalam batas $\frac{1}{2}$ LSB dari nilai akhirnya. Jangka waktu ini tergantung pada kapasitas liar (*stray capacitance*), waktu tunda saturasi dan lain – lain. Waktu pemantapan ini sangat penting karena harganya

menentukan batas kecepatan dalam mengubah berbagai *input*-an digital, biasanya harganya sekitar nano detik sampai mikro detik.

2.5 ADC 0804

ADC (*Analog To Digital Converter*) fungsinya sebagai media untuk mengkonversikan sinyal analog dari alat, plant, ataupun sensor untuk dirubah menjadi sinyal digital sehingga dapat dibaca oleh mikrokontroler ataupun langsung oleh PPI untuk diolah lebih lanjut oleh komputer.



Gambar 2.12. ADC 0804

Fungsi kaki – kaki pada ADC0804 :

1. CS : Sinyal untuk mengaktifkan komponen
2. RD : Sinyal kontrol untuk mengambil data
3. WR : Sinyal kontrol untuk memulai awal konversi
4. CLK – IN : Untuk mengatur besarnya *clock internal*.
5. INT : Status untuk mengetahui bahwa konversi sudah selesai.
6. Vin (-) : *Input*-an analog negatif.

7. Vin (+) : *Input-an analog positif.*
8. A – GND : *Analog ground.*
9. Vref/2 : *Setengah tegangan referensi untuk skala penuh .*
10. D-GND : *Dihubungkan ke ground.*
11. – 18. LSB - MSB : *Data 8 bit.*
19. CKL – R : *Untuk mengatur besarnya clock internal.*
20. VCC : *Tegangan catu daya*

Komponen ini menggunakan pendekatan (*Succesive approximation*) berturut – turut untuk mengkonversi *input-an analog* dalam jangka waktu tertentu menjadi data digital 8 bit. ADC 0804 mempunyai pembangkit pulsa internal dengan sedikit tambahan komponen eksternal, membutuhkan tegangan sebesar +5V, dan mempunyai waktu konversi optimum sekitar 100 mikro detik.

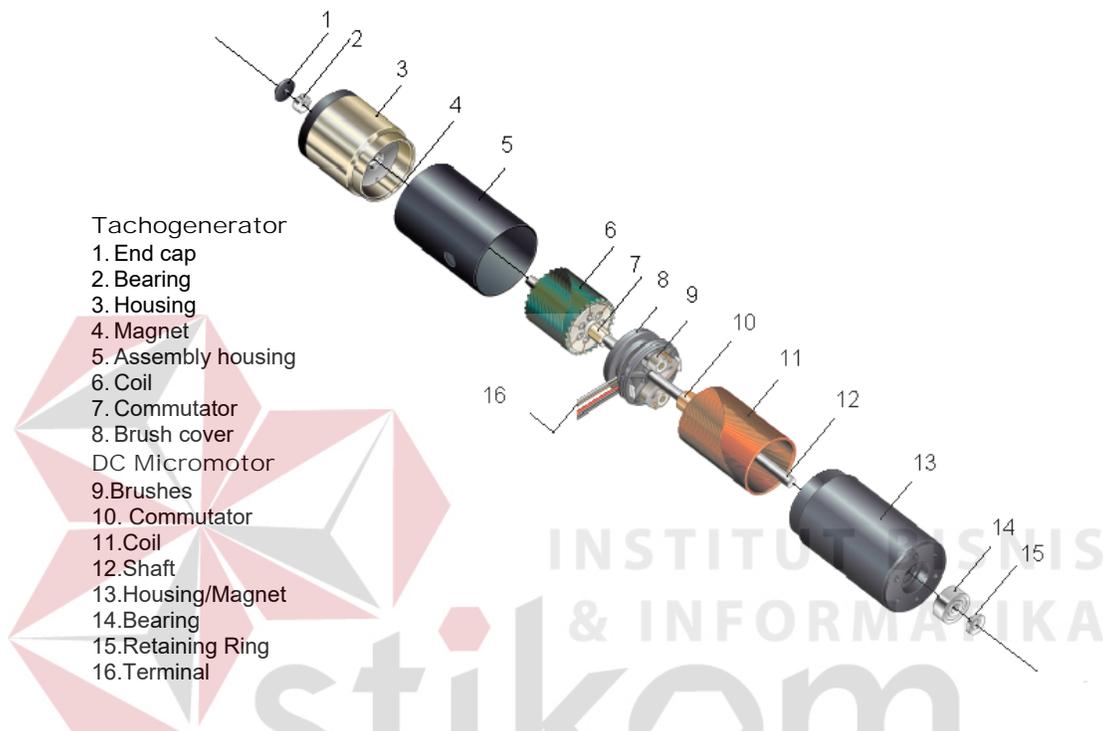
Tegangan *input* maksimum yang dapat dikonversikan bergantung pada seberapa besar tegangan pada pin Vref/2 dengan tegangan absolut maksimum sebesar 18V. Apabila pin Vref/2 diberi tegangan sebesar 5V yang berarti data digital FF pada *output* bersesuaian dengan data analog *input* sebesar 5V, maka resolusi dari ADC ini adalah 20 mV. Resolusi dalam hal ini adalah perubahan tegangan untuk kenaikan tiap bitnya adalah 20mV, didapat dari rumus :

$$resolusi = \frac{V_{ref}}{255} \quad (2.13)$$

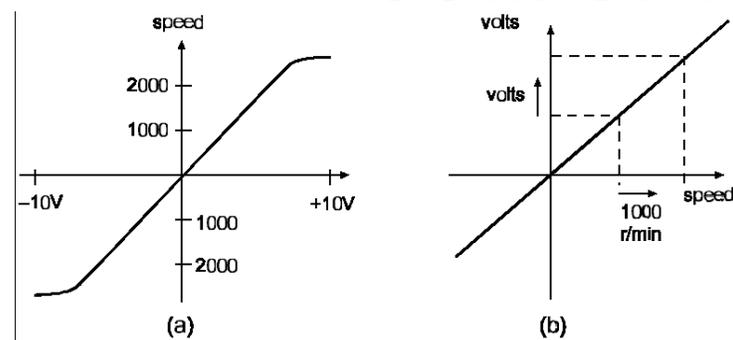
2.6 Tachogenerator

Tachogenerator adalah suatu alat yang bisa merubah energi gerak (mekanis) menjadi energi listrik. *Tachogenerator* biasanya sudah terhubung pada bagian atas Motor DC, hal ini memudahkan pengguna karena tidak harus

menggunakan alat yang banyak untuk mengetahui *output* Motor DC tersebut, tetapi *output* dari *Tachogenerator* tersebut masih merupakan tegangan AC, sehingga diperlukan Rangkaian Penyearah untuk mendapatkan tegangan DC.



Gambar 2.13 Bagian – bagian dari *Tachogenerator*.



Gambar 2.14. Karakteristik dari (a) Motor DC dan (b) *Tachogenerator*

Menurut Servo Fundamentals Trainer untuk *Tachogenerator* yang umum digunakan untuk menghasilkan sebuah tegangan sebesar 2.5V diperlukan kurang lebih 1000 putaran / menit.

2.7 Rangkaian Penyearah Dan Rangkaian Pelemah

Rangkaian penyearah berfungsi untuk merubah tegangan AC menjadi tegangan DC. Rangkaian ini memiliki 2 kelompok :

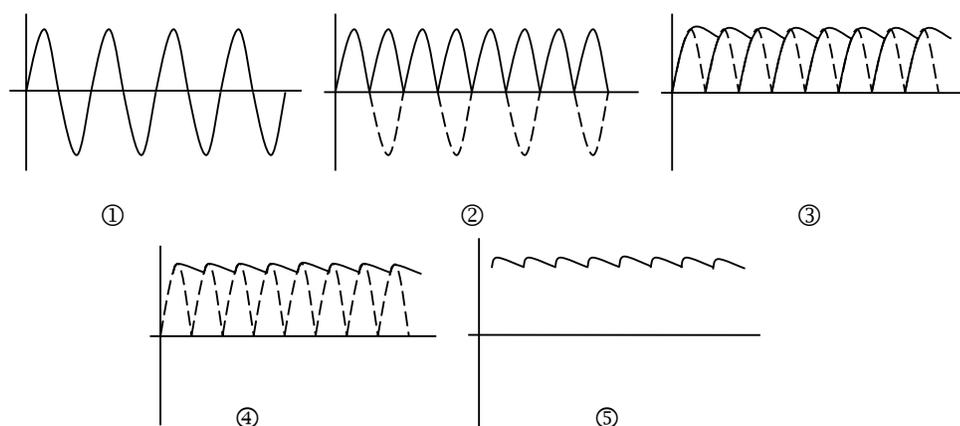
1. Penyearah Setengah Gelombang.

Pada bentuk yang paling sederhana, penyearah setengah gelombang dapat dibuat dengan menghubungkan sebuah Dioda secara seri dengan catu daya AC dan tahanan beban. Disini Dioda hanya mengalirkan arus selama setengah gelombang catu AC.

Kelemahan dari penyearah jenis ini pada tegangan beban, walaupun satu arah, tapi akan berubah – ubah dengan variasi yang tak dapat diabaikan, dan selain itu tegangan beban berharga nol pada setengah perioda. output tegangan DC dari rangkaian penyearah diharapkan serata mungkin, dan di haruskan tegangan beban tidak nol pada saat setengah gelombang sumber AC. Untuk menghilangkan nilai nol tersebut bisa digunakan sebuah Kapasitor yang di pasang secara paralel dengan beban, sehingga aliran arus akan memberi muatan pada Kapasitor, dan tegangan pada Kapasitor akan bertambah. Selama Dioda tak mengalirkan arus, Kapasitor akan melepaskan muatan melalui Resistor beban dan mencegah tegangan beban turun menjadi nol, sehingga di dapat tegangan DC yang mendekati garis lurus.

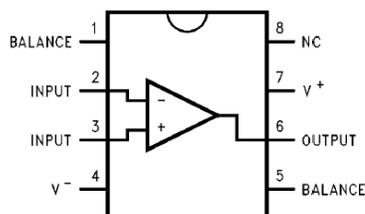
2. Penyearah Gelombang Penuh.

Untuk golongan ini sumber AC akan memanfaatkan kedua setengah gelombang, positif maupun negatif. Setengah gelombang negatif akan terbalik, sehingga di dapat gelombang satu arah. Disini di butuhkan 2 Dioda serta Kapasitor. Keuntungan yang didapat adalah lebih efisien, hanya sedikit menimbulkan magnetisasi DC, dan tegangan kerutnya mempunyai frekuensi dua kali frekuensi catu. Metode lainnya dengan menggunakan Rangkaian Jembatan (*Diode Bridge*), disini dibutuhkan 4 Dioda, Resistor dan Kapasitor. Gambar 2.15 dibawah menunjukkan proses perubahan tegangan AC ke tegangan DC dengan metode penyearah gelombang penuh sehingga setengah gelombang negatif akan terbalik, sehingga di dapat gelombang satu arah. Kapasitor sebagai media penyimpan tegangan sementara waktu gelombang positif naik, sedangkan waktu gelombang turun Kapasitor melepaskan tegangan yang disimpannya sehingga gelombang tidak turun sepenuhnya. Sehingga bentuk gelombang akan menyerupai garis lurus dengan riak kecil.



Gambar 2.15 Proses perubahan tegangan AC ke tegangan DC

Rangkaian Pelemah berfungsi untuk melemahkan tegangan sesuai dengan yang kita kehendaki, rangkaiannya dapat menggunakan 2 buah Op-Amp (*Operasional Amplifier*) serta beberapa R. Untuk rangkaian ini digunakan Op-Amp tipe LF351 atau UA 741.



Gambar 2.16. Op-Amp LF351

Fungsi Pin – Pin Op-Amp LF351 :

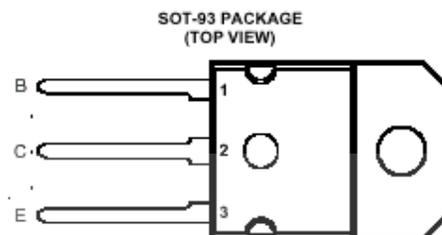
1. *Balance*, tidak digunakan atau *No Connect*.
2. *Input -*, sebagai *input-an* negatif Op-Amp.
3. *Input +*, sebagai *input-an* positif Op-Amp.
4. *Vee*, dihubungkan dengan tegangan -12V DC.
5. *Balance*, tidak digunakan atau *No Connect*.
6. *Output*, sebagai hasil output dari Op-Amp.
7. *Vcc*, dihubungkan dengan tegangan +12 DC.
8. *NC*, tidak dihubungkan (*No Connect*).

2.8 Driver Motor DC

Driver Motor DC digunakan untuk memperkuat arus hasil *output* dari DAC. DAC memiliki *output* arus hanya beberapa milli Ampere, sedangkan kebutuhan Motor DC terkadang lebih dari 3 Ampere, sehingga apabila tidak

menggunakan *Driver Motor DC*, DAC yang digunakan bisa rusak dikarenakan beban arus yang diperlukan untuk Motor DC tidak dapat dipenuhi.

Disini digunakan TIP3055 yang dapat menguatkan arus menjadi mencapai 15 Ampere, sehingga dapat mencukupi beban arus yang dibutuhkan oleh Motor dan juga melindungi DAC dari kerusakan arus balik dari Motor.



Gambar 2.17 TIP 3055

2.9 Borland Turbo C++

C++ diciptakan satu dekade setelah C oleh Bjarne Stroustrup, Laboratorium Bell, AT&T, pada tahun 1983. Bahasa ini bersifat kompatibel dengan bahasa pendahulunya C. Pada mulanya C++ disebut sebagai “*a better C*” kemudian oleh Rick Mascitti pada musim panas 1983 diganti sebagai C++, adapaun tanda ++ berasal dari nama operator penaikan pada bahasa C.

Keistimewaan C++ adalah mendukung untuk pemrograman berorientasi obyek. Tujuan utama pembuatan C++ adalah untuk meningkatkan produktifitas pemrogram dalam membuat aplikasi. Kekompleksitasan bahasa pemrograman C++, terutama pada program yang besar yang terdiri dari 10.000 baris atau lebih.

Borland Turbo C++ merupakan bahasa pemrograman yang sering digunakan untuk belajar bahasa pemrograman secara dasar. C++ selanjutnya

menjadi dasar dari pemrograman tingkat tinggi yang banyak sekali bermunculan, seperti Java, Perl, Python, dan lain sebagainya. Bahasa C++ dari Borland ini mendukung untuk mode secara grafik, walaupun untuk mode grafik masih belum maksimal seperti pemrograman tingkat tinggi sekarang.

Borland C++ juga mendukung operasi I/O yang dapat digunakan dalam dunia kontrol, untuk penggunaannya juga sangatlah mudah. Alat I/O yang didukung C++ dan sering digunakan adalah PPI dan PCL. Untuk operasi *database* sederhana C++ sangatlah handal untuk digunakan, dan disinilah peran pemrograman berorientasi obyek berfungsi.

2.10 MATLAB

MATLAB (Matrix Laboratory) merupakan sebuah *software* yang canggih untuk komputasi teknik. Didalamnya terdapat kemampuan penghitungan, visualisasi, dan pemrograman dalam suatu lingkungan yang mudah untuk digunakan karena permasalahan dan pemecahannya dinyatakan dalam notasi matematika biasa. Secara umum kegunaan MATLAB untuk :

1. Matematika dan komputasi.
2. Pengembangan algoritma.
3. Pemodelan, simulasi dan pembuatan *prototype*.
4. Analisa data, eksplorasi dan visualisasi.
5. Pembuatan aplikasi, termasuk pembuatan antarmuka grafis.

Dari semua kemampuan MATLAB di atas maka *software* ini sangatlah bagus untuk digunakan dalam sistem kontrol. MATLAB juga dapat beroperasi pada sistem operasi Unix, Linux ataupun Windows.