

# PENGONTROLAN KECEPATAN DAN ARAH PUTARAN MOTOR SERVO DC DENGAN BANTUAN PERSONAL KOMPUTER (PC) DAN INTERFACE

Oleh :  
Haruman Wiranegara  
Jurusan Teknik Mesin -Fakultas Teknik UNJANI

## ABSTRAK

*Personal Computer* (PC) yang sekarang ini banyak digunakan untuk kebutuhan pengetikan atau penggambaran, bisa juga digunakan untuk keperluan pengendalian sebuah mesin, namun PC tersebut tidak dapat langsung digunakan, akan tetapi harus dilengkapi dengan perangkat-perangkat lain yang dikhususkan untuk mengendalikan mesin tersebut, kelengkapan itu antara lain adalah *hardware interface*, *software*, dan beberapa kelengkapan lain berupa komponen-komponen pada mesin yang dikendalikan oleh PC. Motor servo DC adalah salah satu penggerak yang banyak sekali digunakan pada sebuah mesin, untuk pengontrolan kecepatannya dapat digunakan PC sebagai perangkat pengendali.

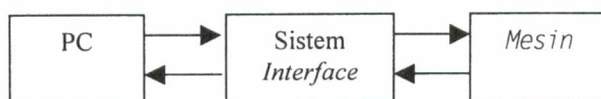
## Pendahuluan

Untuk kebutuhan pengontrolan dengan PC ada dua bagian utama yang harus dilengkapi pada mesin tersebut yaitu : PC dan *Interface*. Untuk PC secara mudah kita peroleh, namun untuk *interface* kita harus merancang dahulu sesuai dengan spesifikasi yang diharapkan.

Dalam disiplin ilmu rekayasa bahasan ini bisa dikembangkan lebih jauh dikarenakan peralihan dari mesin-mesin/instrumen yang digerakan secara manual menjadi dapat digerakan dan dikendalikan oleh PC atas dasar pengetahuan perancangan *interface*, dan juga bisa menjadi satu langkah yang lebih maju dalam disiplin ilmu rekayasa.

## Interface Sebagai Perantara Antara PC dan Mesin.

Pertama kita lihat beberapa hal yang utama dalam pemahaman tentang *Interface* (antarmuka) sebagai penghubung antara PC dan mesin. Sebelumnya akan dilihat secara umum mengenai proses-proses pengendalian mesin dengan PC dilihat melalui diagram blok. Berikut ini adalah diagram blok proses masukan dan keluaran dari sebuah mesin yang dilengkapi dengan PC



## Gambar 1. Diagram blok masukan dan keluaran dalam sistem mesin dan *Personal Computer* (PC)

Pada gambar 1. secara umum terlihat bahwa antara PC, sistem *interface*, dan mesin terdapat tanda anak panah yang menunjukkan adanya suatu proses masukan dan keluaran antar sistem alat, bila dilihat dari gambar diatas yang menjadi penengah atau pembatas antara PC dengan mesin adalah *interface*. Pada umumnya suatu mesin yang menggunakan PC pada proses kerjanya pada sistem alat itu terjadi adanya perubahan sinyal, pada gambar diatas peranan *interface* secara umum menjadi suatu fungsi perantara, disanalah terjadinya proses yang dinamakan perubahan sinyal.

Seorang operator mesin, menggunakan mesin yang dilengkapi dengan PC, akan mengoperasikan mesin dengan cara menjalankan PC dan membuat suatu instruksi-instruksi melalui sebuah *software* bahasa pemrograman. Bahasa pemrograman itulah yang nanti menjadi sebuah instruksi-instruksi. Instruksi-instruksi tersebut diolah dalam PC sedemikian rupa menjadi sinyal-sinyal tegangan digital.

Sinyal-sinyal tersebut kemudian dikirim dan diteruskan melalui *interface* untuk

diubah menjadi sinyal tegangan listrik analog, dimisalkan untuk menggerakkan komponen penggerak mesin tersebut memerlukan tegangan analog (Misalkan untuk motor servo DC).

Pada umumnya sinyal tegangan analog yang masih lemah untuk menggerakkan komponen penggerak yang berdaya besar, maka sebelumnya harus diperkuat dahulu tegangannya dengan penguat sinyal, kemudian baru bisa dikonsumsi tegangannya untuk menggerakkan motor DC sehingga semua transmisi mesin menjadi bisa bergerak.

Pergerakan tersebut menghasilkan respon posisi dan respon kecepatan aktual pada bagian yang diukur, dimana sensor-sensor ditempatkan. Akibat dari gerakan sensor karena transmisi yang bergerak mengakibatkan perubahan posisi dan kecepatan, dari sensor-sensor itu menghasilkan sinyal-sinyal tegangan listrik, sinyal-sinyal tersebut bisa berupa sinyal analog atau berupa pulsa-pulsa tergantung dari jenis sensor yang dipergunakan.

Sinyal-sinyal yang berupa sinyal analog ataupun pulsa-pulsa tersebut diteruskan melalui *interface* kembali. Masuknya sinyal analog dari sensor-sensor tersebut harus diubah kembali menjadi sinyal tegangan listrik digital, supaya bisa diproses atau dibaca oleh PC.

Adapun sinyal yang berupa pulsa-pulsa tetap harus melalui *interface* karena sinyal tersebut harus dihitung terlebih dahulu melalui sebuah sistem komponen penghitung yang dinamakan *counter*.

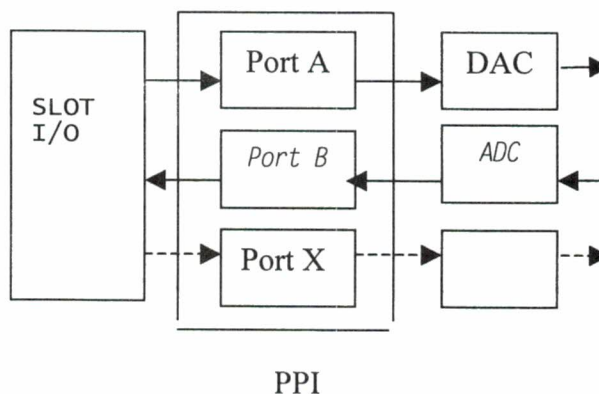
Sinyal-sinyal dari *interface* yang sudah berupa sinyal-sinyal digital, kemudian masuk dan diproses didalam PC. Proses tersebut berlangsung terus-menerus selama waktu pengoperasian mesin.

Hasil proses (*input*) didalam PC, dalam monitor PC akan terlihat berupa angka-angka desimal yang berubah-ubah. Angka-angka tersebutlah yang bisa dijadikan pegangan suatu posisi aktual dari transmisi tersebut, yang sebelumnya telah disesuaikan antara jumlah angka yang diperoleh dengan posisi dan kecepatan yang dicapai.

Dari uraian diatas jelaslah bahwa *interface* menjadi satu sistem penghubung atau yang menjembatani antara mesin dengan PC.

### Diagram Blok *Interface* Pada Perancangan.

Untuk perancangan maka sebaiknya diperhatikan diagram blok komponen dalam sistem *interface*. Pada gambar 2. diperlihatkan bahwa slot ekspansi (I/O) yang menjadi penghubung langsung dengan *interface*. Pada kondisi keluaran sinyal dari slot ekspansi, sinyal tersebut masuk ke sebuah sistem yang berfungsi menuntun jalur yang dipilih instruksi program. Sistem itu dinamakan PPI (*programmable peripheral interface*). Pada PPI tersebut terjadi pengalamatan jalur-jalur port, instruksi yang keluar dari slot ekspansi kemudian dilewatkan pada salah satu jalur port yang dipilih misalnya port A bila jalur Port A dibuat sebagai jalur keluaran dari *interface* sebagai fungsi pengubah sinyal digital-



Gambar 2. Diagram blok rangkaian *interface*

ke analog, sinyal dari port A diubah menjadi analog melalui DAC (*digital to analog converter*) kemudian sinyal dari DAC diperkuat oleh *Op. Amp.* (*operational amplifier*), sebelum tegangan dipakai oleh komponen mesin yang memerlukan konsumsi tegangan yang lebih besar.

Pada kondisi masukan, yaitu sinyal-sinyal tegangan listrik yang masuk dari

komponen sensor masih relatif kecil, pada umumnya harus diperkuat dahulu oleh sistem penguat. Sinyal analog diteruskan untuk diubah menjadi sinyal digital bila jalur masukan itu sebagai jalur pengubah sinyal dari analog ke digital yaitu melalui ADC (*analog to digital converter*). Sinyal yang berupa digital dari jalur tersebut, kemudian diteruskan ke PPI kembali, dengan maksud agar dapat dipanggil oleh instruksi programnya sesuai dengan alamat portnya, kemudian diteruskan kedalam slot ekspansi.

### Slot Ekspansi Pada PC

Untuk dapat membuat modul tambahan pada IBM-PC, kita harus mengetahui sinyal-sinyal yang tersedia dan harus digunakan. Pada IBM-PC, penambahan modul dapat digunakan dengan memasang modul yang ingin ditambahkan pada slot ekspansi. Berikut adalah nama-nama sinyal yang tersedia pada slot IBM-PC.

- A0-A19, *Output*, adalah alamat 20 bit dari mikroprosesor 8088.
- D0-D7, *Input/Output*, Delapan bit data dua arah dari mikroprosesor 8088.
- MEMR (*Memori Read*), *Output*, Sinyal ini merupakan keluaran dari bus *controller* (8288) yang menunjukkan operasi pembacaan memori.
- MEMW (*Memori Write*), *Output*, Sinyal ini merupakan keluaran dari bus *controller* (8288) yang menunjukkan operasi penulisan memori.
- IOR (*Input Output Read*), *Output*, Keluaran dari bus *controller* yang menunjukkan operasi pembacaan I/O.
- IOW (*Input Output Write*), *Output*, Keluaran dari bus *controller* yang menunjukkan operasi penulisan I/O.
- ALE (*Adress Latch Enable*), *Output*, Sinyal ini merupakan keluaran aktif tinggi yang menunjukkan bahwa alamat A0-A19 saat ini valid (bila ALE = 1). ALE pada IBM PC ini tidak dapat digunakan untuk *dimultipleks* alamat D0-D7 yang berasal dari slot ekspansi. Karena D0-D7 sudah tidak *dimultipleks* alamat.

- OSC (*Oscillator*), *Output*, Keluaran dari 8284A yang mempunyai frekuensi 14.31818 MHz.
- CLK (*Clock*), Keluaran dari 8284A yang digunakan untuk *clock* mikroprosesor.
- RESET DRV (*Reset Drive*), *Output*, Sinyal ini akan aktif ketika komputer pertama kali dihidupkan atau ditekan tombol reset.
- IRQ2-IRQ7, *Input*, Sinyal masukan yang dihubungkan langsung ke 8259 untuk mengaktifkan INTR pada 8088.
- DRQ1-DRQ3 (*DMA Request*), *Input*, Tiga masukan aktif tinggi yang digunakan untuk meminta siklus DMA (8237).
- DACK0-DACK3 (*DMA Acknowledge*), *Output*, Suatu sinyal jabat tangan dari DMA yang mengatakan telah diterimanya sinyal DRQ.
- I/O CH CK (*I/O Channel Check*), *Input*, Sinyal masukan aktif rendah yang memberikan indikasi kesalahan paritas pada data di memori atau I/O.
- I/O CH RDY (*I/O Channel Ready*), *Input*, Sinyal masukan aktif terendah pada pena ini akan memperpanjang siklus memori atau I/O (*wait state*). Sinyal ini tidak boleh dibuat lebih 10 siklus *clock*.
- AEN (*Adress Enable*), *Output*, Jika pena ini berlogika 1, berarti bus saat ini adalah bus DMA. Sedangkan bila berlogika 0 bus saat ini bus mikroprosesor.
- TC (*Terminal Count*), *Output*, Sinyal ini jika berlogika 1 menunjukkan bahwa DMA telah menjangkau terminal pada proses DMA.
- Power dan GND (+5V,+12V,-12V,15V,GND), Sinyal daya dan ground.

### PPI 8255A Komponen Pendukung Pada Perancangan Interface.

Pada bahasa pemrograman mengenal adanya penulisan di alamat (*write*) atau pembacaan di alamat (*read*), port adalah nama jalur alamat tertentu yang terdapat dalam PPI, alamat-alamat tersebut berfungsi

sebagai jalur yang dipilih oleh sinyal masukan (*input*) atau keluaran (*output*) sehingga pada saat proses penulisan di-alamat tertentu, PPI bertugas meneruskan ke alamat yang dimaksud untuk ditulis.

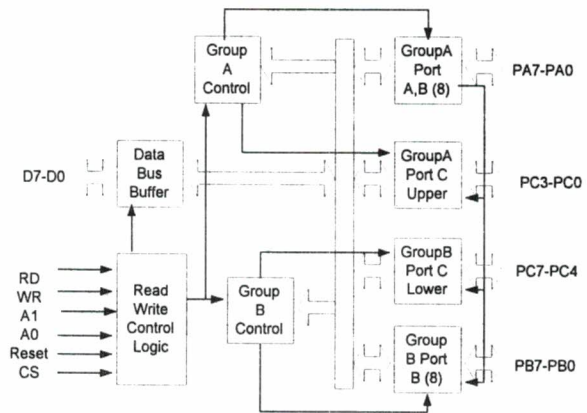
Pada saat proses pembacaan alamat, PPI bertugas menghubungkan alamat yang dimaksud untuk dibaca. Proses penulisan dan pembacaan ini adalah sebagai instruksi mengirim data atau menerima data. Untuk pembahasan masalah penulisan dan pembacaan akan dibahas berikutnya.

Salah satu PPI yang bisa dipakai dalam perancangan adalah Intel PPI 8255A, merupakan piranti *interface* I/O yang dapat diprogram untuk penggunaan dengan sistem komputer. Disebut *programmable* karena fungsinya dapat diatur berdasarkan perangkat lunak. Sebuah kata kendali yang dikirim ke piranti 8255A dapat mengatur 8255A melakukan mode operasi tertentu untuk melaksanakan tugas *interface* yang diperlukan.

Blok diagram dasar dari 8255A ditunjukkan oleh gambar 3. Piranti 8255A dihubungkan dengan sistem komputer melalui bus data dua arah (*bidirectional*) dan dikontrol oleh blok *read/write control logic* yang memiliki masukan-masukan kontrol sebagai berikut:

- CS : *Chip Select*, aktif rendah. Jika pin ini dalam keadaan tinggi, chip tidak dapat diakses (*disable*). Sebaliknya chip baru bisa diakses jika pin ini dalam keadaan rendah.
- RD : *Read*, saat pin ini aktif (pada keadaan rendah), menandakan adanya proses pembacaan data oleh komputer dari bus data. Pin ini dihubungkan dengan pin IOR pada slot PC.
- WR : *Write*, saat pin ini aktif (pada keadaan rendah), menandakan adanya proses penulisan data oleh komputer dari bus data. Pin ini dihubungkan dengan pin IOW pada slot PC.
- A0 dan A1: *Port select* 0 dan 1, menentukan akses ke salah satu dari tiga *register* kontrol yang terdapat pada PPI 8255A.

- Reset : Aktif tinggi, jika pin ini dalam kondisi tinggi, maka chip akan berada dalam keadaan reset.



Gambar 3. Diagram blok piranti 8255A intel.

Port I/O 8255A dibagi ke dalam tiga group, yaitu group A, B, dan C. Setiap group terdiri dari 8 bit, yang dapat dikonfigurasi oleh perangkat lunak dengan menggunakan kata kendali 8 bit.

PPI 8255A memiliki tiga mode operasi, yaitu mode 0, mode 1, dan mode 2. Mode 0 merupakan dasar mode operasi *input/output* yang memungkinkan operasi *input/output* satu arah untuk setiap port tanpa jabatan.

Seperti yang telah dijelaskan diatas bahwa komponen PPI ini berfungsi untuk meneruskan jalur-jalur alamat masukan dan keluaran ke jalur alamat yang dimaksud, dalam hal ini kita harus menentukan satu komponen PPI yang akan kita pergunakan. Dalam perancangan ini penulis memilih PPI8255A sebuah komponen PPI untuk kebutuhan merancang *interface* yang relatif banyak orang yang mempergunakan komponen ini serta mudah untuk memperolehnya karena banyak dijual di pasaran.

Untuk menggunakan dan memfungsikan komponen PPI 8255A ini dalam rangkaian antarmuka seorang perancang harus terlebih dahulu memahami sifat-sifat dari komponen ini melalui buku petunjuk yang menerangkan tentang komponen-komponen PPI

PPI 8255A merupakan piranti I/O serbaguna yang terdiri dari port A (1 byte), port B (1 byte), dan port C (4 bit port C atas dan 4 bit port C bawah). PPI 8255A dapat diprogram menyerupai tiga buah port I/O sederhana (mode 0), dua port I/O jabat tangan (mode 1), atau satu port I/O *bidirectional* dengan lima sinyal jabat tangan (mode 2). Mode-mode tersebut juga bisa digunakan secara bercampur (*intermixed*). Sebagai contoh, port A dapat diprogram untuk beroperasi dalam mode 2 sementara port B beroperasi dalam mode 0. Disamping itu juga terdapat mode bit *set/reset* yang memungkinkan bit-bit dari port C untuk di set atau reset untuk tujuan pengontrolan. Konfigurasi dan nama-nama pin PPI 8255A dapat dilihat pada tabel 1.

**Tabel 1. Konfigurasi nama pin PPI 8255A**

D7-D0	BUS DATA (2 ARAH)
RESET	MASUKAN RESET
CS	CHIP SELECT
RD	MASUKAN BACA
WR	MASUKAN TULIS
A0-A1	PORT ALAMAT
PA7-PA0	PORT A(BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
VCC	+5 VOLT
GND	0 VOLT

Pemrograman PPI 8255A dapat dilakukan secara mudah, yaitu dengan memberikan sebuah kata kendali ke port kendali. Alamat untuk masing-masing port dapat dilihat pada tabel 2. (tentang operasi dasar 8255A).

**Tabel 2. Operasi dasar PPI 8255A**

A1	A0	RD	WR	CS	Operasi masukan
0	0	0	1	0	Bus data=port A
0	1	0	1	0	Bus data=port B
1	0	0	1	0	Bus data=port C
					Operasi keluaran
0	0	1	0	0	Bus data=port A
0	1	1	0	0	Bus data = port B
1	0	1	0	0	Bus data=port C
1	1	1	0	0	Bus data = kendali
					Fungsi disable

A1	A0	RD	WR	CS	Operasi masukan
X	X	X	X	0	Bus data=tiga keadaan
1	1	0	1	0	Kondisi ilegal
X	X	1	1	0	Bus data=tiga keadaan

### DAC (*Digital to Analog Converter*).

Pada sistem elektronik banyak sekali alat-alat yang membutuhkan tegangan analog, sementara keluaran dari komputer merupakan tegangan digital. Untuk dapat mengendalikan alat lain yang membutuhkan tegangan analog, komputer memerlukan suatu penghubung dinamakan DAC (*digital to analog converter*). Gambar 4. adalah salah satu contoh IC DAC yang rangkaian internalnya menggunakan metode tangga R-2R. Nama-nama sinyal pada IC tersebut adalah :

- GND : Merupakan penghubung ground.
- $V_{EE}$  : Harus dihubungkan ke tegangan negatif  $-5V$  sampai  $+15 V$ .
- $I_{out}$  : Arus hasil konversi sinyal masukan.
- D7-D0 :Masukan data digital yang akan dikonversi.
- $V_{cc}$  : Dihubungkan ke sumber tegangan  $+5V$ .
- $V_{ref(-)}$  dan  $V_{ref(+)}$  : Tegangan referensi positif dan negatif.
- Kompensasi dihubungkan pada kapasitor untuk komponen frekwensi.

Gambar 4 adalah salah satu contoh penetapan IC DAC 0808 untuk menghasilkan tegangan keluaran yang sebanding dengan masukan digital yang diberikan.

Sinyal keluaran pada pena 4 merupakan arus yang besarnya tergantung pada nilai D7 sampai D0 dan arus referensi. Hubungan arus keluaran terhadap D7 sampai D0 adalah :

$$I_0 = V_{ref}/R_{14} \times R_0 (D7/2 + D6/4 + D5/8 + D4/16 + D3/32 + D2/64 + D1/128 + D0/256)$$

Arus referensi ( $V_{ref} / R_{14}$ ) biasanya diatur agar bernilai 2 mA. Arus keluaran pada pena 4 dihubungkan ke rangkaian penguat inverting yang akan mempunyai tegangan keluaran sebesar :

$$V_0 = -I_f.R_0$$

Sehingga tegangan DAC akan mempunyai rumus

$$V_0 = I_0 \times R_0$$

$$= (V_{ref}/R_{14}) \times R_0 (D7/2 + D6/4 + D5/8 + D4/16 + D3/32 + D2/64 + D1/128 + D0/256)$$

Mari kita coba menghitung beberapa tegangan untuk arus referensi 2mA dan  $R_0=5k\Omega$ . Untuk data 1000 0000B didapat.

$$V_0 = 2mA \times 5k\Omega (1/2+0+0+0+0+0+0+0) = 5V$$

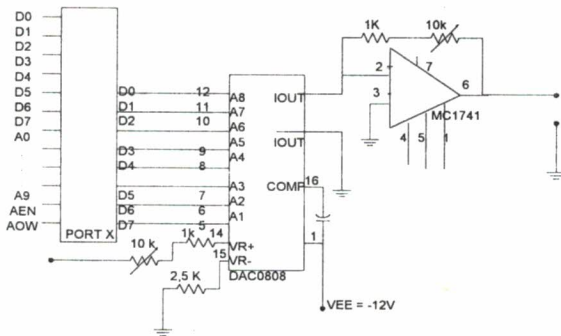
Jadi, pada bit MSB akan dihasilkan tegangan 5V. Untuk data 00000001B didapat.

$$V_0 = 1 mA \times 5k (0 + 0 + 0 + 0 + 0 + 0 + 0 + 01/256) = 10/256 = 39mA$$

Jadi, pada LSB akan dihasilkan tegangan 39 mV. Bila seluruh bit bernilai 1(1111111B) didapat.

$$V_0 = 2 \times 5k. (1/2 + 1/4 + 1/8 + 1/16 + 1/32 + 1/64 + 1/128 + 1/256) = 10 \times 255/256 = 10 \text{ volt}$$

(pada prakteknya adalah 9.961V)



**Gambar 4. Perantara DAC pada port output**

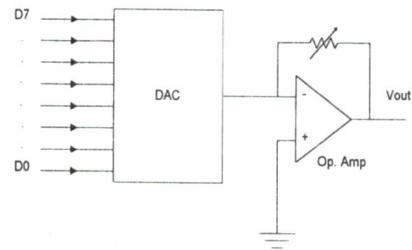
#### DAC 1408L.

Pada keperluan tertentu dalam rangkaian elektronik kadang diharuskan mengubah sinyal dari sinyal digital menjadi sinyal analog dikarenakan dua kondisi pengolahan yang berbeda, untuk itu diperlukan sistem komponen DAC. DAC adalah sebuah sistem komponen yang berfungsi mengubah sinyal digital menjadi sinyal analog, dimana sinyal tegangan listrik keluaran dari komputer yang berupa digital harus diubah menjadi sinyal analog, misalkan untuk fungsi memutar motor penggerak. Untuk keperluan pada sebuah transmisi mesin DAC digunakan

pada jalur keluaran yang berfungsi sebagai jalur instruksi memutar motor servo DC.

Fungsi dari pengubah sinyal digital ke analog adalah mengubah data biner (digital) menjadi sinyal tegangan analog yang sebanding. Gambar 5 memperlihatkan sebuah rangkaian DAC dengan penguat operasional sebagai pengubah arus ke tegangan.

Hal yang perlu diperhatikan pada DAC adalah resolusi, tegangan keluaran skala penuh dengan waktu konversi. Resolusi merupakan spesifikasi terpenting dari DAC. Resolusi DAC bergantung pada jumlah bit-bit masukannya. Sebuah DAC 8 bit seperti gambar 5 memiliki  $2^8$  atau 256 tingkat keluaran yang mungkin, sehingga resolusinya adalah sebesar  $1/256$  atau 0,39%.



**Gambar 5. komponen DAC**

Tegangan keluaran skala penuh dari DAC pada gambar diatas adalah tegangan yang keluar dari pin 6 penguat operasional jika seluruh bit masukan berlogika satu. Keluaran DAC pada skala penuh tidak akan sama dengan keluaran yang dikehendaki, yaitu akan berkurang sebesar  $1/256$ -nya.

Waktu konversi yaitu waktu yang dibutuhkan untuk menyelesaikan satu konversi sinyal digital menjadi analog.

Fungsi utama unit ini adalah untuk mengubah data digital yang dikirim oleh komputer menjadi data tegangan analog, sehingga data tersebut dapat digunakan untuk menggerakkan motor. Unit ini terdiri dari rangkaian DAC, buffer, dan latch data.

Komponen DAC yang digunakan yaitu DAC 1408L produksi motorola. DAC ini juga didukung oleh beberapa komponen lain

seperti penguat operasional, resistor dan kapasitor.

Buffer yang digunakan adalah jenis SN74125. Buffer ini diletakan diantara data digital yang dikirim ke DAC dengan terminal masukan digital DAC. Buffer ini berfungsi untuk menyangga data yang lewat sehingga data digital yang dikirim ke DAC tidak mengalami perubahan.

Chip SN 74125 berisi 4 buah *buffer*. Tabel kebenarannya adalah seperti yang ditunjukkan oleh tabel 3. *latch data* yang digunakan adalah jenis MC 14508 yang diletakan dimuka buffer SN74125. Tujuan pemasangan *latch* ini adalah untuk menahan dan menyimpan data lewat pada saat saluran kontrol dalam keadaan logika '0', Sedangkan dalam keadaan logika '1', data yang melewati latch ini diteruskan. Dengan adanya latch ini, maka perubahan *input/output* dari rangkaian *interface PPI 8255A* tidak mempengaruhi keadaan DAC 1408.

Tabel 3. Kebenaran buffer.

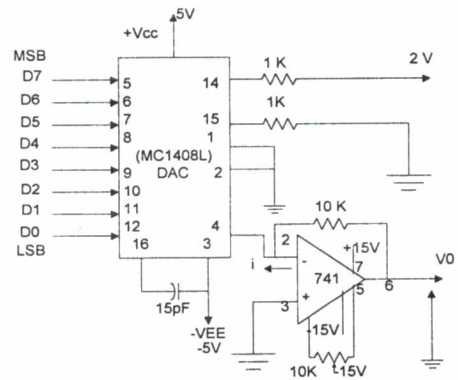
A	C	Y
0	0	0
1	0	1
0	1	Impedansi Tinggi
1	1	Impedansi Rendah

Gambar rangkaian piranti MC14508 adalah seperti yang ditunjukkan oleh gambar 6. Merupakan DAC 8 bit dengan sebuah keluaran untuk arus yang kemudian diubah menjadi tegangan dengan menggunakan sebuah penguat operasional pengubah arus ke tegangan.

Unit pengubah tegangan digital ke analog dirancang agar dapat menghasilkan tegangan keluaran analog positif dan negatif. Tujuannya agar unit ini dapat menggerakkan motor dalam dua arah, yaitu arah CW dan arah CCW. Dengan demikian sebelum keluaran DAC dihubungkan dengan penguat servo, keluaran DAC harus dihubungkan dulu dengan sebuah penguat diferensial. Setelah keluaran DAC dihubungkan dengan penguat diferensial maka tegangan

keluarannya dapat diatur berharga positif dan negatif untuk setiap kata digital tertentu yang dikirim ke DAC

Penguat diferensial yang dirancang diharapkan menghasilkan tegangan keluaran positif maksimum jika seluruh masukannya dalam keadaan logika '1' dan menghasilkan tegangan keluaran negatif maksimum jika seluruh keluarannya dalam keadaan logika '0' Sedangkan pada saat masukan DAC berupa data digital 128, keluaran dari penguat diferensial diharapkan berharga nol volt. Harga nol volt ini nantinya digunakan untuk menghentikan motor. Rangkaian penguat diferensial yang dirancang oleh seperti ditunjukkan oleh gambar.



Gambar 6. DAC 8-bit 1408L

Diagram blok keseluruhan unit pengubah tegangan digital ke analog yang direncanakan adalah seperti ditunjukkan oleh gambar 6.

Jika harga tegangan keluaran penguat diferensial yang diharapkan adalah  $\pm 1$  volt. Berarti  $V_x$  harus berharga 1 volt dan  $V_0$  harus berharga 0V-2V. Hubungan rata  $V_{out}$  dan  $V_0$  dapat dinyatakan sebagai berikut :

$$V_{out} = (V_0 - V_x) \text{ Volt}$$

Dimana :

$V_0$  = Tegangan keluaran dari DAC

$V_x$  = Tegangan referensi.

$V_{out}$  = Tegangan keluaran penguat diferensial.

Jika  $V_0 < V_x$ , maka  $V_{out} < 0$  ; Jika  $V_0 = V_x$ , maka  $V_{out} = 0$

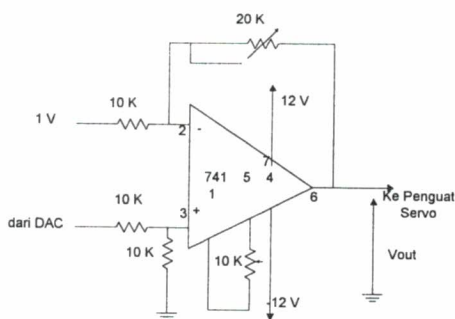
Rangkaian lengkap unit pengubah tegangan digital ke analog yang dirancang adalah seperti ditunjukkan oleh gambar 6. Keluaran unit ini nantinya langsung dihubungkan kepada masukan unit penguat servo.

Untuk keperluan ini kita harus memilih jalur untuk operasi penulisan, pada perancangan ini penulis memilih jalur PA0 s/d PA7 untuk keperluan perubahan sinyal digital ke analog. Kita memerlukan sebuah komponen DAC, untuk keperluan ini penulis menggunakan IC DAC 0808 atau MC 1408. Perhatikan gambar 6.

### Instruksi Penulisan Alamat (*Write*).

Dalam bahasa pemrograman mengenal adanya istilah penulisan alamat dan pembacaan alamat, dimana instruksi penulisan dan pembacaan inilah yang nantinya akan menjadi suatu fungsi yang bermaksud memberikan instruksi atau menerima data antara PC dengan mesin

Tahap selanjutnya setelah memiliki perangkat keras *interface* maka kita harus



Gambar 7. Penguat diferensial op amp. 741

membuat program komputernya. Dalam pembuatan program komputer kita tidak lepas dari rangkaian yang telah dibuat dikarenakan alamat-alamat yang direncanakan sebelumnya harus diaplikasikan pada program yang kita buat.

Program komputer yang akan dibuat adalah dengan mempergunakan perangkat lunak (*software*) bahasa pemrograman Turbo Pascal Versi 7.00 dimana Turbo Pascal memiliki fasilitas untuk keperluan

ini. Dalam pembuatan program kita harus terlebih dahulu mempelajari lebih dalam mengenai pembuatan program-program dalam turbo pascal ditambah dengan mempelajari instruksi penulisan (*write*) dan instruksi pembacaan (*read*), dikarenakan dalam kebutuhan ini tidak hanya memerlukan teknik penulisan atau pemanggilan alamat saja melainkan kita juga harus dapat mengolah tampilan layar komputer, pengolahan rumus-rumus dan lain-lain, itu semua harus dikuasai terlebih dahulu.

Instruksi penulisan alamat artinya bahwa pada perangkat lunak bahasa pemrograman dibuat sebuah instruksi untuk mengirim data berupa angka heksadesimal ke alamat di PPI *interface*.

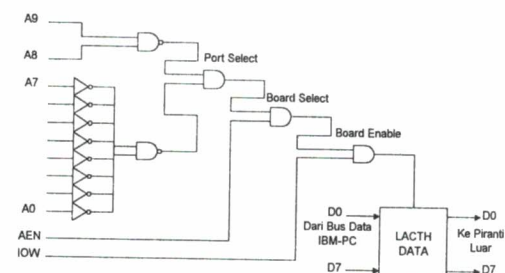
Alamat yang dimaksud kemudian di hubungkan dalam PPI sehingga terjadilah penyaluran sinyal melalui jalur pada alamat tersebut. Fungsi dari alamat ini adalah untuk pemilihan jalur keluaran didalam sistem *interface* kedalam komponen mesin.

Penulisan di alamat pada umumnya bermaksud untuk memberikan instruksi dari PC ke mesin, misalkan untuk memerintah agar motor berputar selama lima menit. Maka dengan instruksi yang dibuat sedemikian rupa dibuat dalam program komputer menghasilkan putaran motor selama lima menit.

Saat komputer akan melaksanakan operasi tulis, misalnya akan mengeluarkan data bilangan 81 H dengan alamat port keluaran 300H, instruksi bahasa pascalnya adalah:

**Port[\$300]:=81;**

Diagram blok dari rangkaian operasi tulis dapat dilihat pada gambar 8.



Gambar 8. Rangkaian operasi tulis



Kondisi jalur alamat, sinyal AEN, IOW, dan data adalah seperti ditunjukkan oleh tabel berikut:

A9	A8	A7	A6	A5	A4
1	1	0	0	0	0
A3	A2	A1	A0	AEN	
0	0	0	1	0	
D5	D4	D3	D2	D1	D0
0	0	0	0	0	1

Karena kondisi *board write strobe* aktif, maka data akan keluar. Jika instruksi mengeluarkan data setelah selesai, *board write strobe* tak aktif dan jalur data akan tertahan pada kondisi bilangan 81 H sampai operasi tulis berikutnya diperintahkan.

### Uji Jalur DAC

Pengujian selanjutnya yaitu pengujian jalur DAC. Pengujian dilakukan dengan cara melakukan pengukuran tegangan *output* dari jalur DAC dengan *multitester*. Pengujian ini bertujuan mengetahui apakah data I/O dari PPI 8255A telah tersalurkan dengan baik terhadap *Latch data*, *buffer* dan DAC.

Pengujian dilakukan dengan cara menjalankan program *write* pada alamat 300H dengan nilai *output* yang diubah-ubah. Bila pada *multitester* diketahui terjadinya perubahan tegangan analog *high-low* maka dapat dipastikan bahwa jalur DAC kita berfungsi dengan baik dan bila tidak terjadi perubahan *high-low* maka jalur DAC belum berfungsi. Pemeriksaan dilakukan dengan cara memeriksa kembali jalur data D0-D7 dari slot ekspansi, *Latch data*, *buffer* dan DAC.

### Setting Sinyal Digital Ke Analog.

Tegangan keluaran dari keseluruhan unit ini diharapkan berharga antara -10V dan +10V. Data digital 255 yang dikirim ke port DAC diharapkan menghasilkan tegangan keluaran analog maksimum +10, V. Selanjutnya data digital 254 s/d 128 diharapkan akan menghasilkan tegangan analog yang harganya terus berkurang secara linier hingga berharga 0V saat data

digital yang dikirim ke port DAC berada pada harga 128. Sedangkan saat data digital 128 s/d 0 akan menghasilkan tegangan analog yang semakin negatif hingga akhirnya berharga -10 V saat data digital yang dikirim ke port DAC berada pada harga 0. Bagian dari unit pengubah tegangan digital ke analog yang harus dikalibrasi terlebih dahulu adalah rangkaian DAC dan rangkaian penguat differensial.

Pertama-tama yang harus diperiksa saat mengkalibrasi rangkaian DAC adalah tegangan offset yang mungkin timbul pada *op-amp* yang digunakan. Jika ternyata terjadi tegangan offset, maka tegangan offset tersebut dapat dihilangkan dengan memutar VR2 sampai diperoleh tegangan offset sama dengan nol atau mendekati nol. Selanjutnya besarnya penguatan tegangan yang diinginkan dapat diatur dengan cara memutar *variable resistor* VR1. DAC diharapkan supaya menghasilkan tegangan keluaran analog yang harganya berkisar antara 0-2V. Tegangan keluaran analog 0V dihasilkan saat masukan DAC memperoleh masukan data digital 0, sehingga VR1 harus diatur/diputar supaya saat masukan digital DAC berupa data digital 0, keluaran analog DAC berharga 0V dan saat masukan digital DAC berupa data digital 255, keluaran analog DAC berharga maksimum 2V. Sedangkan VR2 berfungsi untuk menghilangkan tegangan offset yang mungkin timbul pada *op-amp* 1.

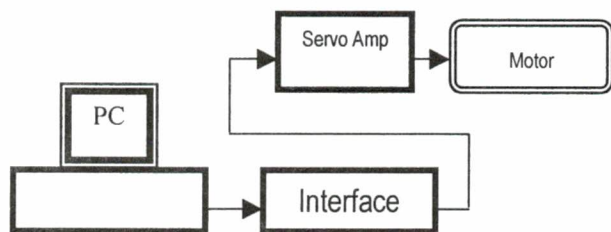
Penguat differensial pada unit ini berfungsi untuk menghasilkan tegangan keluaran yang harganya berkisar antara +10V dan -10V. Pertama-tama yang harus dilakukan pada penguat ini adalah memeriksa apakah pada *op-amp* yang digunakan terdapat tegangan offset. Jika pada *op-amp* timbul tegangan offset, maka tegangan offset tersebut harus dihilangkan dengan cara memutar variabel resistor VR5 sampai diperoleh tegangan offset sama dengan nol atau mendekati nol. Selanjutnya yang harus dilakukan adalah mengatur penguatan tegangan penguat differensial dengan cara memutar variabel resistor VR3 dan VR4. Penguat ini diharapkan supaya menghasilkan tegangan analog maksimum

+0,5 V saat masukan digital pada DAC berupa data digital 255. Kemudian saat masukan digital pada DAC berupa data digital 0, Keluaran penguat ini diharapkan berada pada tegangan minimum -10V. Sedangkan pada saat masukan digital DAC berupa data digital 128, keluaran penguat differensial ini diharapkan berharga 0V.

Pada kalibrasi yang telah dilakukan untuk unit pengubah tegangan digital ke analog. Kalibrasi data dihentikan setelah diperoleh tegangan keluaran maksimum mendekati harga +10V dan tegangan keluaran minimum mendekati harga sebesar -10 V. Hasil ini sudah memadai karena sudah mendekati harga yang diinginkan dan sudah dapat digunakan untuk menggerakkan motor. Setelah dikalibrasi selesai dilakukan, maka berarti unit ini siap untuk menerima perintah dari komputer dan juga siap untuk dihubungkan ke penguat servo. Perlu kita ketahui bahwa tidak selalu kita harus mensetting harga keluaran DAC sampai maksimum yang diterima *servo controller* namun pada prakteknya jika mengalami kesulitan dalam mensetting DAC maka kita cukup berorientasi pada kebutuhan.

### Pemrograman Putaran Motor

Berikut ini kita lihat diagram blok penempatan sistem pengontrolan motor servo DC pada gambar 9.



Gambar 9. Pengontrolan putaran motor.

Pemberian instruksi pada program turbo pascal V-7 untuk menggerakkan motor adalah sebagai berikut :

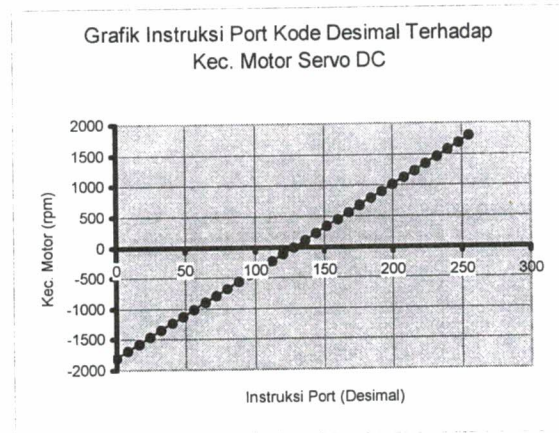
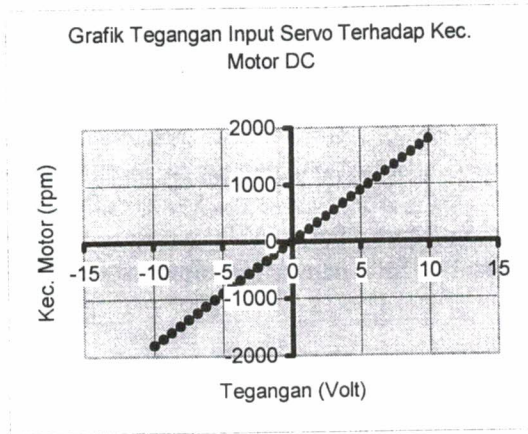
**PORT[\$300]:={0 s/d 255}**

{0 s/d 255} adalah nilai yang dapat diberikan pada program dan efek dari perubahan nilai tersebut adalah variasi putaran dan arah putaran yang berbeda. Sebagai contoh : PORT[\$300]:=128 adalah kecepatan 0 (motor tidak berputar). Pada tabel 4 adalah hasil pengujian motor servo DC terhadap instruksi yang diberikan program Turbo Pascal V-7.

**Tabel 4 Hasil pengujian motor servo DC**

INSTRUKSI PROGRAM TURBO PASCAL V-7	KODE (BINER)	KODE (DESIMAL)	TEGANGAN INPUT SERVO CONTROLLER (VOLT)	KEC. PUTAR MOTOR SERVO DC (RPM)
PORT[\$300]:=\$0	0	0	-10	-1800
PORT[\$300]:=\$8	1000	8	-9.375	-1687.5
PORT[\$300]:=\$10	10000	16	-8.75	-1575
PORT[\$300]:=\$18	11000	24	-8.125	-1462.5
PORT[\$300]:=\$20	100000	32	-7.5	-1350
PORT[\$300]:=\$28	101000	40	-6.875	-1237.5
PORT[\$300]:=\$30	110000	48	-6.25	-1125
PORT[\$300]:=\$38	111000	56	-5.625	-1012.5
PORT[\$300]:=\$40	1000000	64	-5	-900
PORT[\$300]:=\$48	1001000	72	-4.375	-787.5
PORT[\$300]:=\$50	1010000	80	-3.75	-675
PORT[\$300]:=\$58	1011000	88	-3.125	-562.5
PORT[\$300]:=\$60	1100000	96	-2.5	-450
PORT[\$300]:=\$68	1101000	104	-1.875	-337.5
PORT[\$300]:=\$70	1110000	112	-1.25	-225
PORT[\$300]:=\$78	1111000	120	-0.625	-112.5
PORT[\$300]:=\$80	10000000	128	0	0
PORT[\$300]:=\$88	10001000	136	0.625	112.5
PORT[\$300]:=\$90	10010000	144	1.25	225
PORT[\$300]:=\$98	10011000	152	1.875	337.5
PORT[\$300]:=\$A0	10100000	160	2.5	450
PORT[\$300]:=\$A8	10101000	168	3.125	562.5
PORT[\$300]:=\$B0	10110000	176	3.75	675
PORT[\$300]:=\$B8	10110000	184	4.375	787.5
PORT[\$300]:=\$C0	10111000	192	5	900
PORT[\$300]:=\$C8	11000000	200	5.625	1012.5
PORT[\$300]:=\$D0	11001000	208	6.25	1125
PORT[\$300]:=\$D8	11011000	216	6.875	1237.5
PORT[\$300]:=\$E0	11100000	224	7.5	1350
PORT[\$300]:=\$E8	11101000	232	8.125	1462.5
PORT[\$300]:=\$F0	11110000	240	8.75	1575
PORT[\$300]:=\$F8	11111000	248	9.375	1687.5
PORT[\$300]:=\$FF	11111111	255	10	1800

## List Program Turbo Pascal V.7



```
PROGRAM
Pengujian_Motor_servo_DC;
USES CRT;
VAR
I :integer;
BEGIN
I :=0;
clrscr;
repeat
INC(I);
PORT[$303]:=128;
PORT[$300]:=10;
WRITELN('MOTOR BERPUTAR 5 detik
kekiri');
delay(1000);
port[$300]:=1;
clrscr;
until I=1;
end.
```

### Daftar Pustaka

1. Rizal Rizkiawan, 1997. Tutorial Perancangan Hardware 1, Elex Media Komputindo, Jakarta.
2. Rizal Rizkiawan, 1997. Tutorial Perancangan Hardware 2, Elex Media Komputindo, Jakarta.
3. Rizal Rizkiawan, 1997. Tutorial Perancangan Hardware 3, Elex Media Komputindo, Jakarta.
4. Douglas V. Hall, \_Microprocessors and Interfacing, Programming and Hardware, McGraw-Hill
5. Dwihono, 1996. Rangkaian Logika, INDAH, Surabaya..
6. Data Sheet Book 4, 1997. Elex Media Komputindo, Jakarta.
7. Ediman Lukito, 1993. Belajar Sendiri Pemrograman Dengan TURBO PASCAL 7.0, Elex Media Komputindo, Jakarta.
8. Rusman Hakim, 1995. Belajar Sendiri Mengenal SISTEM KOMPUTER, Elex Media Komputindo.
9. Buku Manual: Instructions For Use Of Automatic Load-strain Controller Of Shimadzu Autograph AG-A Series
10. Electrical Maintenance Service Manual For Shimadzu Autograph AG-A Series.