

BAB II

LANDASAN TEORI

2.1 . Teori Pascal

2.1.1. Definisi Pascal

Bahasa Pascal merupakan salah satu bahasa pemrograman tingkat tinggi yang powerful. Dibanding dengan bahasa pemrograman tingkat tinggi yang lain, misalnya FORTRAN, BASIC, COBOL, atau PL/I, Pascal tidak hanya merupakan bahasa lain yang memungkinkan penulisan program-program serupa. Pascal adalah bahasa tingkat tinggi (*high level language*) yang orientasinya pada segala tujuan, khususnya untuk memperkenalkan pemrograman yang terstruktur. Jadi Pascal adalah bahasa yang ditujukan untuk membuat program yang terstruktur .

Pascal merupakan bahasa yang paling cepat populer dibandingkan dengan bahasa – bahasa komputer tingkat tinggi lainnya .

Turbo pascal merupakan versi bahasa Pascal yang paling populer dan banyak digunakan . Kompailer Turbo Pascal banyak digemari, karena terutama bersifat interaktif, seperti interpreter saja layaknya, Turbo Pascal adalah *copyright* oleh BORLAND Inc. dan dapat digunakan pada sistem operasi PC-DOS, MS-DOS, CPM-86 dan CP/M-80.

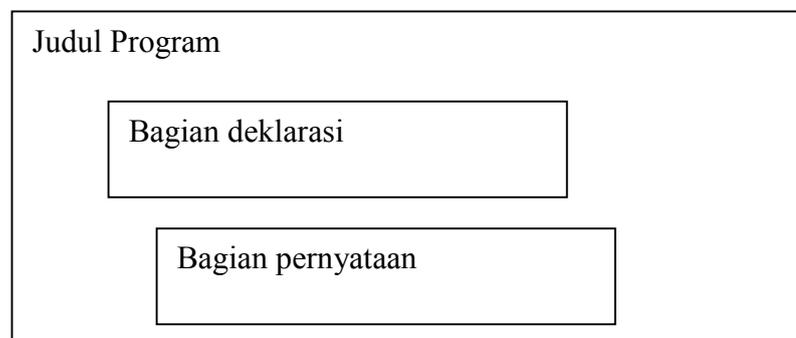
2.1.2. Struktur Program Pascal

Struktur dari suatu program Pascal terdiri dari sebuah judul program (*program heading*) dan suatu blok program (*program blok*) atau badan program (*body program*). Blok program dibagi lagi menjadi dua bagian yaitu bagian

deklarasi (*declaration part*) dan bagian pernyataan (*statement part*). Bagian deklarasi dapat terdiri dari deklarasi label (*labels declation*), deklarasi konstanta (*constants declaration*), deklarasi tipe (*type declaration*), deklarasi variable (*variables declaration*), deklarasi prosedur (*procedures declaration*) dan deklarasi fungsi (*function declaration*) .

Secara ringkas, struktur suatu program Pascal dapat terdiri dari :

- 1) Judul Program
- 2) Blok Program
 - a. *Bagian Deklarasi*
 - deklarasi label (labels declation)
 - deklarasi konstanta (constants declaration)
 - deklarasi tipe (type declaration)
 - deklarasi variable (variables declaration)
 - deklarasi prosedur (procedures declaration)
 - deklarasi fungsi (function declaration)
 - b. *Bagian Pernyataan*



Gambar 2.1. Struktur program Pascal

2.1.3. Program Pascal Yang Paling Sederhana

Suatu program Pascal yang paling sederhana adalah terdiri dari sebuah pernyataan saja. Bagian pernyataan ini menunjukkan suatu tindakan yang akan dikerjakan oleh program. Tindakan yang dilakukan oleh program tergantung dari intruksi – intruksi yang diberikan. Pernyataan atau statemen merupakan intruksi program. Pernyataan – pernyataan yang akan diberikan untuk dikerjakan ditulis diantara kata cadangan “Begin dan End“, Contoh bentuk Program dengan statement kosong :

Begin

End.

Pada bentuk program diatas disebut statemen kosong karena tidak ada statemen sebagai intruksi. Yang perlu diperhatikan adalah setiap statemen perbarisnya diakhiri dengan titik koma (;) kecuali statemen yang terakhir boleh diakhiri dengan titik koma boleh tidak. Untuk menampilkan hasil perbaris digunakan statemen Writeln. Berikut bentuk umum program Pascal dengan statemen:

```
begin
    statemen;
    -
    -
end.
```

Contoh program Pascal yang sederhana :

```
Begin
    Writeln ('Saya pascal');
```

```

Writeln ('.....')
End.

```

Bila program ini dijalankan :

Saya pascal

.....

2.1.4 Mencetak Hasil di Printer

Dalam waktu singkat kita dapat membuat program pascal yang sederhana untuk menampilkan hasil di layar. Bilamana Pascal tersebut ingin ditampilkan di printer, maka dapat menyebutkan nama dari printer tersebut yaitu *Lst*.

Contoh :



Bila program di atas dijalankan hasilnya sbb :

Saya Pascal

Jadi kalau alat tidak disebutkan, maka tampilan akan dilakukan di layar, sedangkan bila ingin ditampilkan I printer, maka alat printer harus disebutkan.

2.1.5. Penulisan Program Pascal

Program pascal tidak mengenal aturan penulisan di kolom tertentu, jadi boleh ditulis mulai kolom berapapun. Penulisan statemen – statemen pada contoh program yang menjorok masuk beberapa kolom tidak mempunyai pengaruh di proses, hanya dimaksudkan supaya mempermudah pembacaan program, sehingga akan lebih terlihat bagian – bagiannya, bisa juga penulisannya disambung dalam satu baris :

```
Begin
```

```
    Writeln ('Saya Pascal');
```

```
    Writeln ('.....')
```

```
End.
```

Atau `Begin writeln ('Saya Pascal'); writeln ('.....') end.`

Tapi penulisan yang langsung disambung seperti di atas *kurang dianjurkan*, dan sebisa mungkin dihindarkan, karena tidak baik untuk dokumentasi program dan sulit untuk membacanya

2.1.6. Judul Program

Judul program sifatnya adalah optional dan tidak signifikan didalam program. Jika ditulis dapat digunakan untuk memberi nama program dan daftar dari parameter tentang komunikasi program dengan lingkungannya yang sifatnya sebagai *dokumentasi* saja. Judul program bila ditulis, harus terletak pada awal program dan diakhiri dengan titik koma.

Contoh :

```
Program contoh ;  
Begin  
Writeln ('saya Pascal');  
Writeln ('.....');  
End.
```

Misalnya program hanya menampilkan hasil saja, tetapi tidak meminta data input, pada judul program dapat didokumentasikan dengan ditulis sebagai berikut :

Program contoh (output);

Atau

Program contoh (layar);

Parameter tentang komunikasi program dengan lingkungan ditulis di dalam tanda kurung buka dan dapat ditulis apapun dengan dipisahkan oleh koma. Contoh penulisan judul program yang benar :

Program Gaji (input, output);

Program Laporan (layar, printer, plotter);

2.1.7. Bagian Deklarasi

Bagian ini digunakan bila di dalam program menggunakan pengenalan (*identifier*). Identifier dapat berupa label, konstanta, tipe, variabel, prosedur dan fungsi. Kalau program menggunakan identifier, Pascal menuntut supaya identifier

tersebut dikenalkan terlebih dahulu sebelum digunakan, yaitu dideklarasikan terlebih dahulu pada bagian ini.

2.1.7.1. Deklarasi Konstanta

Bila kita ingin menggunakan identifier yang berisi nilai – nilai konstanta maka harus didefinisikan terlebih dahulu pada bagian ini. Definisi konstanta diawali dengan kata cadangan *Const* diikuti oleh kumpulan identifier yang diberi suatu nilai konstanta. Ada beberapa konstanta tidak perlu didefinisikan dapat dipergunakan langsung :

- False dengan type Boolean, bernilai logika salah
- True dengan type Boolean, bernilai logika benar
- MaxInt dengan type numerik integer, bernilai 32767.
- MaxLongInt dengan type numerik integer, bernilai 214715.347.

2.1.7.2. Deklarasi Variabel

Jika konstanta merupakan *identifier* berisi data konstanta yang nilainya sudah di tentukan dan pasti, tidak dapat dirubah di dalam program, maka variabel adalah *identifier* yang berisi data yang dapat berubah – ubah nilainya di dalam program. Jadi dengan menggunakan pengenalan konstanta, anda tidak dapat merubah nilainya di dalam program, tetapi dengan menggunakan variabel, nilainya dapat anda ubah – ubah di dalam program. Hampir setiap program pascal yang cukup panjang pasti menggunakan variabel. Setiap kata cadangan *Var* digunakan sebagai judul di dalam bagian deklarasi variabel dan diikuti oleh satu atau lebih *identifier* yang dipisahkan koma, diikuti dengan titik dua dan tipe dari datanya serta diakhiri dengan titik koma.

2.1.7.3. Deklarasi Type

Suatu data yang akan dipergunakan di Program Pascal harus disebutkan typenya. Pascal; menyediakan beberapa macam type dari data, terdiri dari :

a. Data Type Sederhana

Data tipe ini dihubungkan dengan sebuah identifier untuk sebuah data. Simple type dapat digolongkan menjadi type data standart dan type data yang didefinisikan oleh pemakai. Yang termasuk type data standart adalah data dengan tipe *integer*, *real*, *char*, *string* dan *boolean*. Yang termasuk user defined data type adalah *enumerated* atau *scalar* (sebuah data disebutkan satu persatu) dan *subrange* (range dari data).

b. Data Type Terstruktur

Tipe ini terdiri dari beberapa data item yang dihubungkan satu dengan lainnya. Masing – masing grup dari data item dihubungkan dengan suatu identifier tertentu . Ada 4 tipe data yaitu : *array*, *record*, *file* dan *set* .

c. Type Data Penunjuk

Untuk mendefinisikan type dari data yang akan dipergunakan di dalam program, dapat dilakukan dengan mendeklarasikan pengenalan variabelnya di bagian deklarasi *variable* sesuai dengan tipe data yang dipergunakan. Kata cadangan Type digunakan sebagai judul deklarasi type dan diikuti oleh satu atau lebih *definisi type* yang dipisahkan dengan titik koma.

2.1.7.4. Deklarasi Label

Jika suatu program menggunakan statemen *Go to* untuk meloncat ke suatu statemen maka dibutuhkan label pada statemen yang dituju dan label tersebut

harus dideklarasikan dahulu pada bagian deklarasi dengan menggunakan kata cadangan Label diikuti oleh kumpulan identifier label dengan dipisahkan oleh koma dan diakhiri titik koma .

2.1.7.5 Deklarasi Prosedur

Prosedur merupakan bagian yang terpisah dari program dan dapat diaktifkan dimanapun dalam program. Prosedur dapat berupa prosedur standar maupun prosedur yang akan dibuat sendiri oleh pemakai (bila program akan dibagi menjadi blok – blok modul). Kata cadangan Procedure digunakan sebagai judul dari bagian deklarasi.

2.1.7.6. Deklarasi Fungsi

Blok fungsi hampir sama dengan blok procedure, tapi kalau fungsi harus dideklarasikan dengan typenya yang menunjukkan type hasil dari fungsi. Type tersebut ditulis pada akhir deklarasi fungsi yang didahului dengan titik koma :

FUNCTION identifier daftar parameter): type;

2.2. TEORI DASAR ROBOT

Banyak definisi telah dikembangkan untuk membedakan robot dengan perangkat otomatis lainnya. Hal ini sejalan dengan perkembangan teknologi robot itu sendiri. Kebanyakan dari definisi tersebut mempunyai beberapa kesamaan mendasar, yaitu:

- Programable (mampu program)
- Automatic (otomatis)
- Manipulators (perangkat pemanipulasi)
- Humanlike (mempunyai kemiripan dengan manusia)

Robot tidak hanya merupakan mesin perkakas biasa, namun robot adalah mesin khusus yang dikontrol oleh komputer, yang dapat melakukan berbagai pekerjaan dengan menggunakan kemampuan yang mirip dengan kemampuan yang dimiliki oleh manusia. Dua definisi yang telah diterima oleh kalangan industri tentang robot, ditinjau dari berbagai sudut pandang adalah:

1. Dikembangkan oleh ***Computer Aided Manufacturing-International (CAM-I)***. “ Robot adalah peralatan yang mampu melakukan fungsi-fungsi yang biasanya dilakukan oleh manusia, atau peralatan yang mampu bekerja dengan intelegensi yang mirip intelegensi manusia “.
2. Dikembangkan oleh ***Robotics Institute of America (RIA)***, perkumpulan pembuat robot, yang lebih menitik beratkan terhadap kemampuan nyata yang dimiliki oleh robot dari pada kemiripannya dengan manusia, “ Robot adalah peralatan manipulator yang mampu program, mempunyai berbagai fungsi, yang dirancang untuk memindahkan barang, komponen-komponen, peralatan, atau alat-alat khusus, melalui berbagai gerakan terprogram untuk melaksanakan berbagai pekerjaan “.

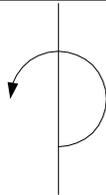
Sedangkan kata robot itu sendiri diambil dari kata *robota* yang berasal dari bahasa Ceko, yang mempunyai arti *pekerjaan*, dipopulerkan oleh Issac Asimov pada tahun 1950 dalam sebuah karya fiksinya. Untuk jaman sekarang, robot industri bukanlah peralatan yang dirancang untuk menirukan keseluruhan kemampuan manusia. Tetapi robot industri lebih menyerupai tangan manusia yang dirancang dengan mencontoh kemampuan gerakannya. Oleh karena itu, robot

industri dikenal pula dengan istilah robot tangan atau umum pula disebut robot manipulator.

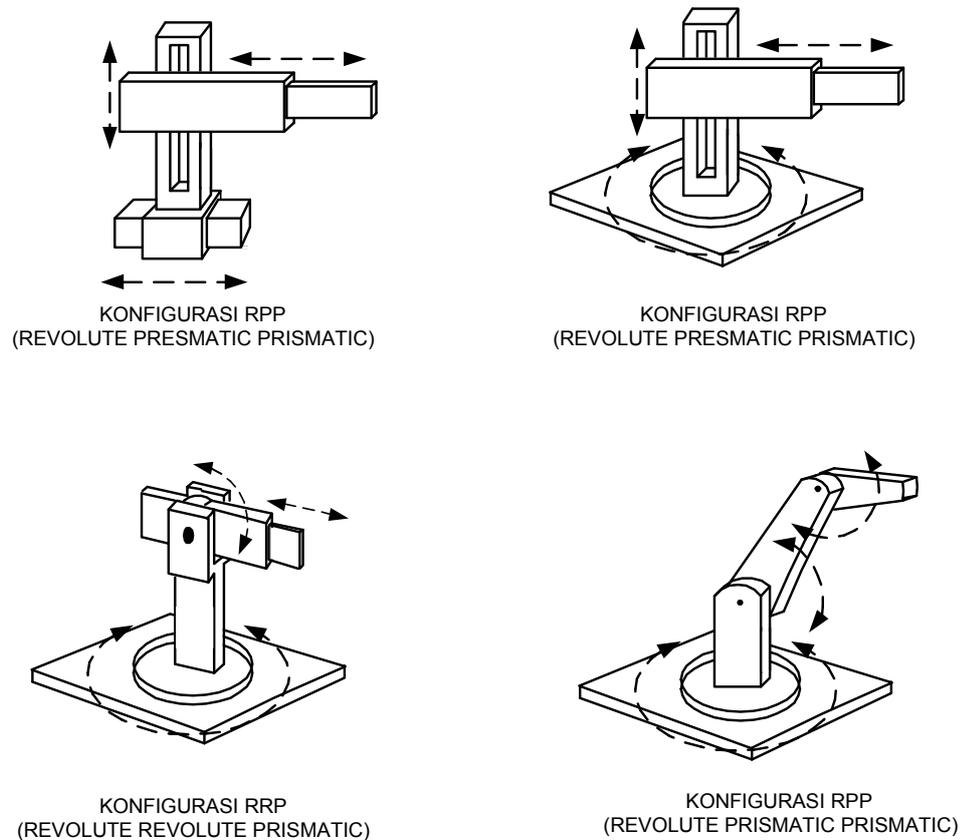
2.2.1 Konfigurasi mekanis

Pada robot industri terdapat dua jenis dasar sambungan antara bagian-bagian robot yang umum dipakai yaitu seperti diperlihatkan pada tabel 2.1.

Tabel 2.1. Jenis Sambungan Bagian Robot

Jenis sambungan	Notasi	Simbul	Keterangan
Revolute	R		Gerakan melingkar terhadap sumbu
Prismatic	P		Gerakan memanjang sepanjang sumbu

Dari tabel 2.1 diatas dapat dibentuk konfigurasi dari sumbu utama robot yang terdiri dari konfigurasi-konfigurasi seperti diperlihatkan pada gambar dibawah ini.



Gambar 2.2 Konfigurasi Mekanis Robot

Beberapa bentuk konfigurasi dari sumbu utama robot :

- **PPP** (*Prismatic, Prismatic, Prismatic*)

Robot dengan konfigurasi ini disebut sebagai *Rectangular Coordinate Robot*.

Ketiga sambungan prismatic bertujuan untuk menggerakkan pergelangan robot keatas dan kebawah, masuk dan keluar, serta maju dan mundur. dari jenis gerakan ini diperoleh lingkup ruang kerja robot yang berbentuk kotak segi empat. Sedangkan jika robot ini dipasang dari atas pada kerangka segi empat maka disebut *gantry robot*.

- **RPP** (*Revolute, Prismatic, Prismatic*). Robot dengan konfigurasi seperti ini disebut sebagai *Cylindrical Coordinate Robot*. Bagian sambungan jenis revolute akan memutar tangan robot, sedangkan sambungan jenis prismatic akan menggerakkan pergelangan robot keatas dan kebawah sepanjang sumbu tegak robot serta bergerak kedepan dan kebelakang sepanjang sumbu radial robot. Oleh karena robot ini mempunyai posisi radial minimum maka lingkup ruang kerjanya akan berupa *silinder konsentris*.
- **RRP** (*Revolute, Revolute, Prismatic*). Disebut juga *Spherical Coordinate Robot*, sambungan revolute yang pertama akan memutar tangan robot kedepan dan kebelakang terhadap sumbu tegak robot, sedangkan sambungan revolute yang kedua akan memutar tangan robot keatas dan kebawah terhadap sumbu horizontal dari pundak robot. Bagian sambungan jenis prismatic akan menggerakkan pergelangan robot kedepan dan ke belakang secara radial. Lingkup ruang kerja yang dihasilkan akan berbentuk seperti bola yang terpotong bagian atas, bawah, dan bagian belakang robot. Hal ini terjadi karena keterbatasan gerak dari sambungan bagian robot.

Dengan konfigurasi RRP dapat dibentuk *SCARA Robot (Selective Compliance Assembly Robot Arm)*. Pada robot semua sumbu utamanya adalah tegak. Lingkup ruang kerjanya menjadi semakin kompleks serta sangat dipengaruhi oleh jangkauan gerak dari sumbu utamanya.

- **RRR** (*Revolute, Revolute, Revolute*), disebut juga *Articulated Coordinate Robot*. Gerakan robot jenis inilah yang paling mirip dengan gerakan tangan manusia. Sambungan revolute yang pertama akan memutar robot kedepan dan

kebelakang. Sambungan revolute yang kedua akan memutar tangan robot keatas dan kebawah terhadap sumbu mendatar dari siku-siku robot. Gerakan robot ini menghasilkan lingkup ruang kerja yang kompleks dan mempunyai pandangan samping mirip dengan bentuk bulan sabit.

2.2.2. Kebebasan Gerak Robot

Kebebasan gerak, yaitu keanekaragaman lintasan yang dapat dilakukan oleh manipulator robot. Kemampuan gerak robot umumnya dinyatakan dengan istilah derajat kebebasan yaitu menyatakan jumlah sumbu yang dimiliki atau jenis jumlah tipe gerakan independen yang dapat dilakukan oleh robot. Oleh karena manipulasi gerakan robot dilakukan dalam ruang tiga dimensi, maka untuk mendapatkan posisi tangan robot pada sembarang posisi dan orientasi diperlukan enam sumbu, yaitu tiga sumbu utama ditambah dengan tiga sumbu minor. Sumbu minornya diperlukan untuk menggerakkan pergelangan robot. Ketiga sumbu minor tersebut adalah :

- **Roll**, gerakan melingkar pada bidang tegak lurus dengan sisi depan ujung tangan robot.
- **Pitch**, gerakan melingkar pada bidang tegak lurus dengan tangan robot.
- **Yaw**, gerakan melingkar pada bidang mendatar dengan tangan robot.

Sedangkan mekanisme untuk membuka dan menutup jari tangan robot tidak dianggap sebagai sumbu independen, oleh karena gerakan tersebut tidak mempunyai peran dalam menentukan posisi atau orientasi dari pergelangan robot. Untuk robot industri umumnya empat sampai dengan enam derajat kebebasan. Namun hal ini tidak menutup kemungkinan untuk mendapatkan robot yang

mempunyai lebih dari enam derajat kebebasan yang utamanya digunakan untuk mampu mencapai daerah-daerah yang banyak rintangannya.

2.2.3. Sistem Penggerak Robot

Tenaga *Sistem penggerak*, yaitu jenis sumber penggerak yang digunakan untuk menggerakkan manipulator robot. Berdasarkan dari jenis sumber tenaga penggeraknya, penggerak robot dapat di klasifikasikan :

- ***Pneumatic***. Robot ini digerakkan oleh udara bertekanan. Umumnya robot ini berukuran kecil, kurang fleksibel, dan tidak mahal dalam pembuatan dan pengoperasiannya. Kemampuan angkut dan kecepatannya dibatasi oleh tekanan operasi udaranya.
- ***Hydraulic***. Robot ini digerakkan oleh fluida yang dipompakan ke piston silinder, atau bentuk mekanisme penggerak hidrolik lainnya. Umumnya robot ini lebih kompak namun mempunyai kemampuan angkut, tenaga, dan kecepatan yang tinggi serta kemampuan kontrol yang akurat. Oleh karena kemampuan pompa hidroliknya dapat ditempatkan terpisah dari robotnya dan tidak diperlukan sumber listrik untuk menggerakkan manipulatornya maka robot ini lebih aman dan cocok untuk lingkungan basah, berdebu dan yang mudah meledak.
- ***Electric***. Robot ini digerakkan motor listrik yang dapat berbentuk *DC Servo Motor* atau *DC Stepper Motor*, cocok untuk penggunaan dengan kapasitas angkut yang kecil namun diperlukan akurasi yang tinggi. Alat pemegang (*end-effector*) robot umumnya digerakkan oleh pneumatic, khususnya apabila gerakkan yang diperlukan hanya gerakkan membuka dan menutup saja. Oleh

karena udaranya mempunyai sifat mampu kompresi maka proses pengecaman tidak akan merusak benda kerja yang dicekam.

2.2.4. Sistem Kontrol Gerak Robot

Dasar klasifikasi robot lainnya yang sangat penting yaitu berdasarkan pada metode yang digunakan untuk mengontrol gerakan dari alat pemegang (*end-effector*) robot. Berdasarkan kriteria ini dapat diklasifikasikan :

- ***Point to point motion robot.***

Jika *end-effector* robot bergerak pada urutan titik-titik diskrit pada ruang kerja robot. Lintasan antar titik-titiknya tidak dikontrol oleh pemakai robot. Robot jenis ini sangat berguna untuk operasi-operasi yang mempunyai sifat diskrit, sebagai contoh operasi las titik, mengambil dan menempatkan benda kerja dll.

- ***Continuous-part Motion Robot.***

Pada robot ini *end-effector* harus mengikuti lintasan tertentu dalam ruang tiga dimensi dengan kecepatan gerak sepanjang lintasan tersebut dapat berubah. Contoh penggunaannya adalah untuk operasi pengecatan, las kampuh, pengeleman, dll.

2.2.5. Metode Pemrograman Robot

Terdapat tiga metode dasar untuk pemrograman robot yaitu:

- 1) ***Guiding***, atau disebut juga metode “*playback*”.

Metode ini dilakukan dengan jalan menggerakkan tangan robot secara manual yang sesuai dengan gerakan tangan robot yang diinginkan. Selama

gerakan ini sistem kontrol robot akan mencatat data lintasannya, metode ini cukup sederhana, operator robot tidak perlu menulis program robot, namun metode ini hanya cocok untuk gerakan-gerakan yang sederhana dan untuk langkah-langkah yang pendek. Dengan metode ini gerakan awal robot dipandu oleh operator, selanjutnya program yang tersimpan dapat dijalankan kembali. Dalam menjalankan ulang programnya dapat dilakukan perubahan-perubahan kecepatan sesuai dengan yang diinginkan.

2) ***Teach pendant.***

Metode ini menggunakan kontrol panel yang disebut “*teach pendant*” atau “*teaching box*”, yang mempunyai beberapa tombol fungsi untuk menggerakkan robot melalui kabel yang dihubungkan dengan sistem kontrol robot. Dengan “*teaching box*” ini robot dikenalkan pada titik-titik posisi robot yang diharapkan, selanjutnya posisi-posisi tersebut disimpan pada memori komputer. Setelah posisi-posisi yang diinginkan tersimpan pada memori komputer, maka operator robot kemudian membuat program untuk mengontrol keseluruhan gerak robotnya.

3) ***Off-line Programming.***

Metode ini menggunakan bahasa pemrograman tingkat tinggi untuk membuat program yang mengontrol semua gerakan dan aksi dari setiap penggerak bagian robot.

2.3. Input Output dan Slot IBM

2.3.1. Dasar Input Output

Ada dua jenis Komunikasi antara Mikroprosesor dengan Peralatan lain yaitu :

- I/O Mapped (*Peta I/O*) dan Memory Mapped (*Peta Memory*).

Peralatan Disk Drive, Port Printer (*LPT*) dan Port Serial (RS-232) merupakan Peralatan I/O yaitu I/O Mapped, sedangkan Hubungan antara RAM (*Random Access Memory*) dan ROM (*Read Only Memory*) yaitu Memory Mapped.

Dalam proses Input/Output secara I/O Mapped terdapat dua metode yaitu :

- 1). Direct Method (Secara Langsung).

Metode Pengalamatan ini dilakukan secara langsung alamat I/O yang dituju oleh Program.

Alamat I/O yang dapat digunakan / diakses yaitu dari 00h sampai FFh (0 – 255h) atau dibawah 255h (FFh)

- 2). Indirect Method (Secara Tidak Langsung)

Metode Pengalamatan ini dilakukan secara tidak langsung alamat I/O yang dituju oleh Program.

Alamat I/O yang dapat digunakan yaitu 0 – 6553h atau diatas 255h.

Alamat Port dicatat di Register DX, sedangkan data yang dikeluarkan dimasukkan di Register AL (Untuk 8 Bit Data) dan Register AX (Untuk 16 Bit Data)

Pada Metode Indirect menggunakan 16 Bit Alamat yaitu A0 – A15 untuk berhubungan dengan Piranti luar (Alamat 0000h – FFFFh).

Namun Pada Komputer alamat yang dapat digunakan untuk rangkaian Interfacing adalah A0 – A9

2.3.2. Sistem Slot pada IBM PC.

Sistem dari IBM PC terdiri dari tiga bus yaitu *Alamat Bus*, *Data Bus*, dan *Kontrol Bus*. Data bus mempunyai jalur data 8 bit. Seluruh jalur bus pada IBM PC telah diberi buffer. Secara bersama-sama data bus, alamat bus, dan kontrol bus disebut sistem bus. Pada IBM PC terdapat lima buah slot sistem bus.

Sistem bus pada IBM PC merupakan bagian penting dalam perencanaan suatu interface, untuk mendapatkan kemudahan maka IBM PC dilengkapi dengan lima buah slot yang mempunyai 62 pin card. Slot ini berhubungan langsung dengan sistem bus IBM PC. Sistem bus ini memungkinkan untuk menunjang direct memory access (DMA).

IBM PC menyediakan ekspansi slot yang memungkinkan digunakan dalam berbagai sistem interfacing. Slot ini berguna untuk menghubungkan card yang digunakan dalam pengembangan komputer. Konfigurasi slot IBM PC dapat dilihat pada gambar 2.3.

Berikut ini adalah penjelasan fungsi dari masing-masing card dari slot IBM-PC.

- A0 – A19 :

Merupakan jalur output yang berfungsi sebagai jalur alamat. Digunakan oleh sistem komputer untuk menunjuk alamat memory I/O selama operasi memory atau Operasi Read/Write pada I/O.

- D0 – D7 :

Jalur data bus ini bersifat dua arah baik sebagai input ataupun output, pada jalur ini perlu ditambahkan IC buffer untuk memastikan bahwa data benar-benar dalam keadaan valid.

- AEN (Address Enable) :

Jalur ini digunakan untuk mengaktifkan buffer alamat . Jika jalur ini dalam kondisi aktif berarti buffer alamat siap beroperasi dalam penunjukkan alamat pada memory atau I/O.

- IOR, IOW (I/O Read, I/O Write) :

IOR dan IOW digunakan apabila terjadi operasi Read atau Write oleh CPU dengan rangkaian luar. Pin IOR dan IOW dihubungkan langsung dengan pin RD dan WR pada PPI 8255.

- RESET DRV (Reset Driver) :

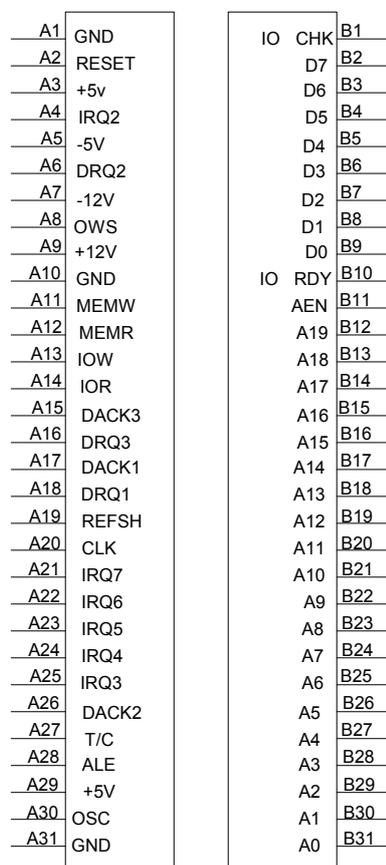
RESET DRV digunakan oleh CPU untuk mereset suatu peralatan luar.

○ POWER BUS dan GROUND :

Untuk mensupply interface card ini diperlukan tegangan dari luar slot. Pin + 5 Vdc, mempunyai toleransi 5% (4,75 V – 5,25 V) pin + 12 Vdc mempunyai toleransi 5% (11,4 V – 12,6 V). Pin –12V mempunyai toleransi 10% (-10,8 V sampai 13,2 V).

Tabel 2.2 Pemetaan Alamat I/O pada IBM PC

ADDRESS	FUNGSI
0000 – 000F	Chip 8292 DMA
0120 – 0021	Chip 8259 Interupt
0040 – 004B	Chip 8255 PPI
0080 – 0089	Register DMA
00A0 – 00AF	Register NMI
00C0 – 00EF	Cadangan 1
0100 – 01FF	Tidak digunakan
0200 – 020F	Game controller
0210 – 0217	Unit Expansi
02FB – 02FF	RS-232C (Skunder)
0300 – 031F	Prototype Card
0320 – 032F	Fixed Disk
0370 – 037F	Printer
0380 – 038C	SDLC Comunication
03A0 – 03A9	Binary Comunication
03B0 – 03BF	Dispaly Monochrome
03C0 – 03CF	Cadangan 4
03D0 – 03DF	Disk Color / Graphic
03E0 – 03E7	Cadangan 5
03F0 – 03F7	Floppy Disk
03F8 – 03FF	RS-232 (Primer)



Gambar 2.3 Konfigurasi Slot IBM-PC

2.3.3. ALAMAT PORT I/O :

Pada Mikroprosesor Intel disediakan 1024 alamat Port untuk I/O (Piranti luar) yang dibagi menjadi 2 bagian :

256 Alamat Port (0000h – 00FFh) untuk System Board (*Motherboard*) dan 768 Alamat Port (0100h – 03FFh) untuk Card Optional atau PPI.

Dari 768 Alamat Port untuk Slot Card Optional dan PPI disediakan alamat (0300h – 03FFh) untuk Slot Card Prototype.

Pengalamatan Card Expansi tidak boleh dilakukan sembarangan untuk menghindari tumpang tindihnya alamat tersebut dengan Piranti lainnya.

Alamat Port pada Komputer :

2F8/IRQ3	:	Serial Port 1
3F8/IRQ4	:	Serial Port 2
378/IRQ7	:	Paralel Port/LPT

Tabel 2.3. Alamat Port pada Slot IBM

Nomor Port	IBM PC XT	IBM PC AT
0000h – 000Fh	Kendali DMA	Kendali DMA
0010h – 001Fh	Undocumented	Reserved
0020h – 002Fh	Kendali Interupsi	Kendali Interupsi 1
0030h – 003Fh	Undocumented	Kendali Interupsi 1
0040h – 005Fh	Timer	Timer
0060h – 006Fh	Port Paralel	Keyboard
0070h – 007Fh	Undocumented	RTC, NMI Mask
0080h – 009Fh	Register halaman DMA	Register halaman DMA
00A0h – 00AFh	NMI Mask Register	Kendali Interupsi 2
00B0h – 00BFh	Undocumented	Kendali Interupsi 2
00C0h – 00CFh	Reserved	Kendali DMA 2
00D0h – 00DFh	Undocumented	Kendali DMA 2
00E0h – 00Efh	Reserved	Reserved
00F0h – 00FFh	Undocumented	Math Coprosesor
0100h – 01Efh	Undocumented	Tersedia untuk chanel I/O
01F0h – 01FFh	Undocumented	Fixed Disk Adapter
0200h – 020Fh	Game I/O Adapter	Game I/O adapter
0210h – 021Fh	Expansion uniter	Reserved
0220h – 024Fh	Reserved	Tersedia untuk channel I/O
0250h – 026Fh	Undocumented	Tersedia untuk channel I/O

0270h – 027Fh	Paralel Printer 2	Paralel Printer 2
0280h – 02Afh	Undocumented	Tersedia untuk channel I/O
02B0h – 02DFh	Alternate EGA	Alternate EGA
02E0h – 02EFh	Undocumented	GPIB 0, akuisisi data 0
02F0h – 02FFh	Secondary asyn. Adapter	Port serial 2
0300 – 031Fh	<i>Prototype Card</i>	<i>Prototype Card</i>
0320h – 032Fh	Fixed Disk Adapter	Tersedia untuk channel I/O
0330h – 035Fh	Undocumented	Tersedia Untuk channel I/O
0360h – 036Fh	Undocumented	Jaringan Komputer
0370h – 037Fh	Paralel Printer 1	Paralel Printer 1
0380h – 038Fh	Second bisyn Controller	Second bisyn controller
0390h – 039Fh	Undocumented	Cluster adapter
03A0h – 03AFh	First bisyn Controller	First bisyn controller
03B0h – 03BFh	Penampil monokrom	Penampil monokrom
03C0h – 03CFh	EGA Adapter	EGA Adapter
03D0h – 03DFh	CGA Adapter	CGA Adapter
03E0h – 03EFh	Reserved	Reserved
03F0h – 03FFh	Floppy Disk Adapter	Floppy Disk Adapter

2.4. Teori Interfacing

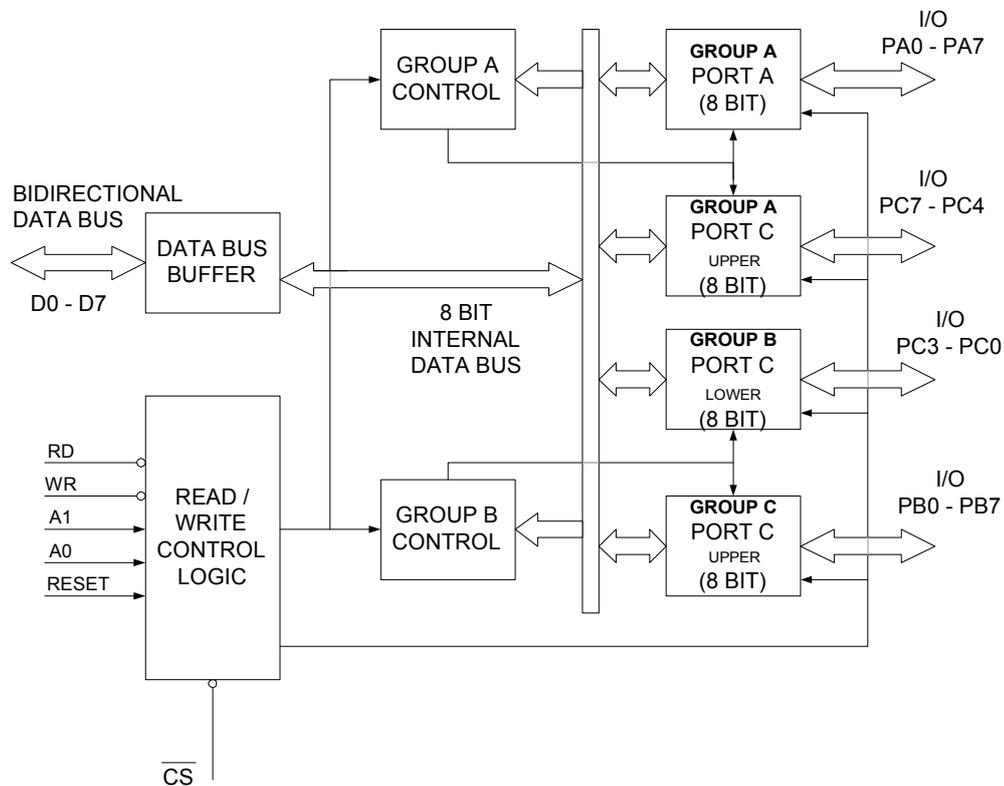
Interfacing adalah suatu perlakuan atau usaha untuk menghubungkan antarmuka / interface dua buah system dapat berupa system intregrasi dari sebuah

IC Pheriperal atau Minimum System. Secara umum pengertian interfacing yaitu perlakuan antara suatu system rangkaian berbasis Mikroprosesor (*Minimum system*) dengan suatu rangkaian luar seperti switch, relay, lampu, rangkaian keyboard, rangkaian monitor, unit printer, rangkaian sensor, dan lain-lain.

PPI 8255 merupakan perangkat Intergated Circuit yang termasuk jenis LSI (*Large Scale Integration*) yang dikemas dalam bentuk 40 pin DIP (*Dual In Package*) dan dirancang untuk melakukan fungsi interfacing dalam sistem komputer, yaitu menghubungkan piranti dari luar sistem komputer dengan sistem dalam komputer .

Disebut *Programable* karena dapat difungsikan sebagai input atau output yang dapat diprogram melalui perangkat lunak (*software*).

PPI 8255 mempunyai tiga buah Port, yaitu port A, port B, dan Port C (PA, PB, dan PC) serta sebuah control word register yang masing-masing terdiri dari 8 bit, dimana dengan menggunakan software masing-masing port (Port A, B dan C) dapat difungsikan sebagai input atau output. Untuk mengetahui cara kerja PPI ini, perlu diketahui blok diagram serta pin-pin yang terhubung pada setiap blok tersebut.



Gambar 2.4. Diagram Blok PPI 8255

Hubungan antara PPI 8255 dengan sistem komputer dapat dilakukan dengan mudah karena PPI 8255 tidak memerlukan sinyal clock, PPI 8255 merupakan I/O, dan bukan memory oleh sebab itu diperlukan sinyal IOR dan IOW yang dihubungkan dengan input RD dan RW pada PPI 8255.

2.4.1. Konfigurasi PPI 8255

Diantara yang perlu diketahui dari PPI 8255 adalah :

- A0 – A1

Kombinasi logika kedua buah input line ini menentukan internal register mana yang akan dituju atau diminta.

- Read / Write dan Control Logic

Fungsi dari blok tersebut adalah mengatur proses perpindahan data didalam PPI 8255 serta proses pengeluaran data ke sistem data bus. Blok tersebut menerima input dari CPU melalui *address bus* dan kontrol bus serta menghasilkan output yang digunakan untuk mengatur blok yang lain dalam PPI 8255.

- CS (*Chip Select*)

Sebagai input enable yang memungkinkan komunikasi (*Proses Read/Write*) antara PPI 8255 dan CPU bila pin CS aktif low.

- RESET

Logika high “1” pada input tersebut menyebabkan PPI 8255 dalam keadaan reset. Hal ini menyebabkan semua port diset sebagai input mode dan control register dihapus (*Clear*). A0 dan A1 kombinasi dari kedua input termasuk menentukan internal register PPI 8255 yang dituju pada proses *Read / Write*.

- D₀ - D₇

Merupakan input / output PPI 8255, semua data diterima dan dikirim melalui jalur ini.

- RD (Read)

Logika Low "0" pada pin ini bersama-sama dengan logika low pada CS memungkinkan untuk mengirimkan data pada sistem data bus pada sistem komputer masuk ke data bus buffer dari PPI 8255.

- Port A, B, C

Merupakan 8 bit I/O port yang dapat diprogram untuk melakukan berbagai macam fungsi. Khusus port C dibagi menjadi 2, yang dapat berfungsi sebagai kontrol sinyal output maupun status input port A dan port B.

Untuk menghubungkan PPI 8255 ke sistem prosesor dapat dilakukan dengan data bus ke data bus lines dari PPI 8255. A0 dan A1 dengan address lines A0 dan A1 dari prosesor. Untuk CS dihubungkan dengan rangkaian IC komparator yang alamatnya sudah ditentukan sebelumnya. Perlu diperhatikan bahwa untuk PPI 8255 ini hanya memerlukan 4 lokasi I/O untuk port A, port B, port C dan *Control Word Register*. Fungsi dari control word register adalah untuk menyimpan kombinasi bit yang mengatur mode kerja dari PPI 8255.

Keempat register ini dapat diakses dengan pengaturan kombinasi dari address A0, A1, WR, RD dan CS yang merupakan input dalam operasi dasar PPI 8255 dan operasi yang terjadi dapat dilihat pada tabel dibawah ini.

Tabel 2.4 Internal Register

A ₁	A ₀	NAMA REGISTER
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control Word

Tabel 2.5 Hubungan antara RD, WR, A₀, dan A₁.

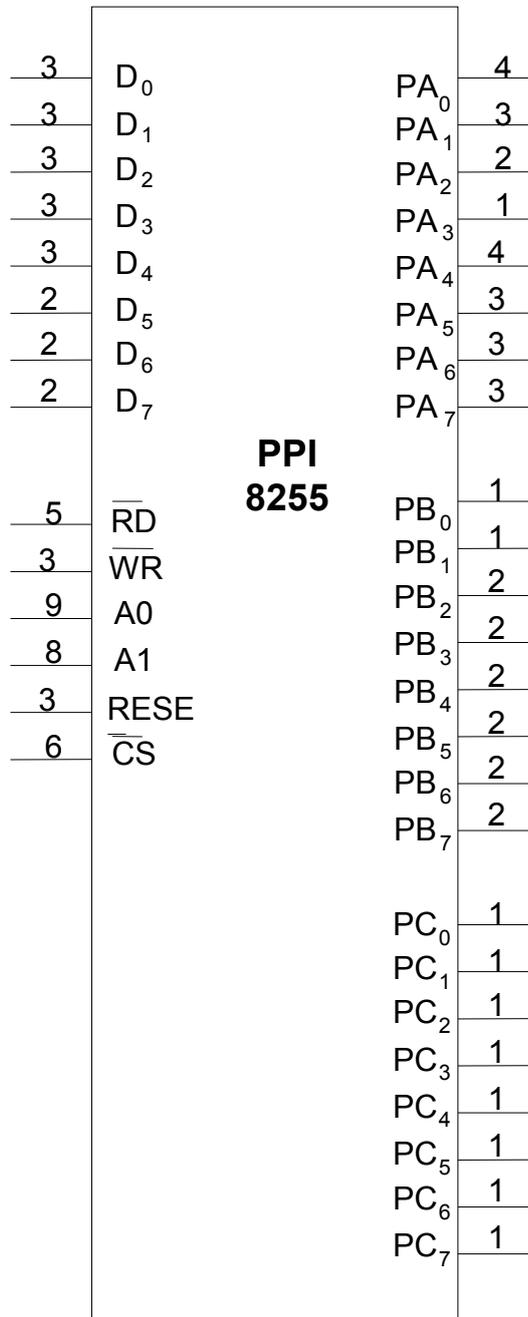
RD	WR	A ₀	A ₁	OPERASI
1	0	0	0	Membaca data port A
0	1	0	0	Menulis data port A
1	0	0	1	Membaca data port B
0	1	0	1	Menulis data port B
1	0	1	0	Membaca data port C
0	1	1	0	Menulis data port C
1	0	1	1	Membaca data Control Word
0	1	1	1	Menulis data Control Word

PPI 8255 mempunyai 3 mode operasi, yaitu :

- Mode 0 : Basic Input / Output
- Mode 1 : Strobe Input / Output
- Mode 2 : Bidirectional Bus.

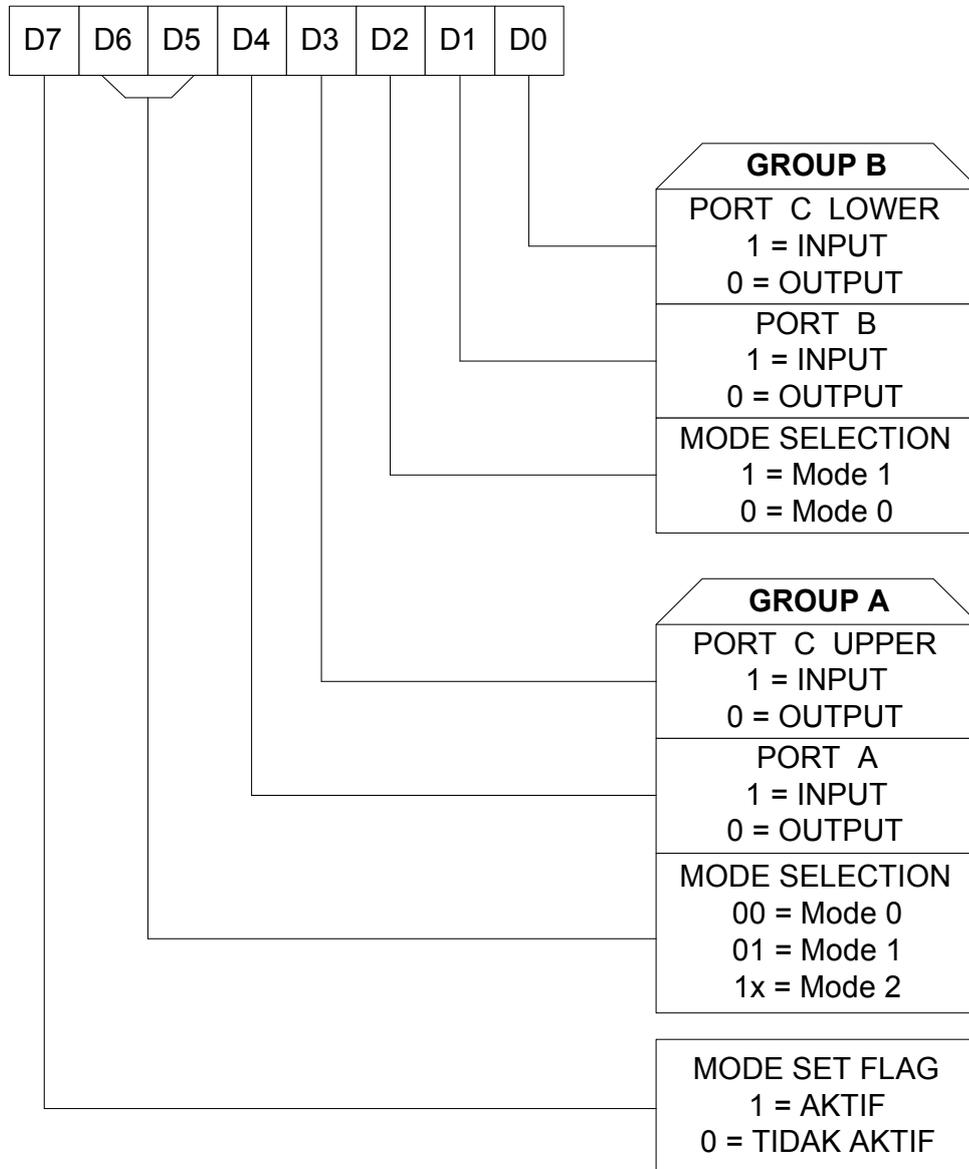
Mode-mode ini dapat dipilih dengan memberikan data tertentu pada *control word* register (*CWR*) saat inisialisasi umumnya diberikan dalam bentuk software.

Control word yang cukup satu kali diberikan, yakni pada saat awal PPI diaktifkan. Selain itu selama program berjalan, PPI 8255 sewaktu-waktu dapat dirubah mode operasinya dengan memberikan data ke *Control Word* lagi. Tersedianya fasilitas ini memungkinkan PPI 8255 dapat dimanfaatkan untuk melayani berbagai keperluan interfacing dengan program-program yang berbeda dalam satu sistem.



Gambar 2.5 Konfigurasi Pin PPI 8255

Tabel 2.6. Inisialisasi PPI 8255



Tabel 2.7. Control Word

A0	A1	RD	WR	CS	MREQ	IOREQ	OUTPUT
1	1	1	0	1	1	0	DATA → CWR

Sebuah *control word* harus dikirim terlebih dahulu ke *control register* melalui perangkat lunak (*software*). *Control word* ini nantinya menentukan kerja masing-masing register. Tabel 2.7 memperlihatkan bahwa:

- Bit D7 menandakan bahwa word ini *control word*.
- Bit D6 dan D5 mendefinisikan mode kerja PPI 8255.
- Bit D4 menunjukkan bahwa port A dioperasikan sebagai input / output port.
- Bit D3 menunjukkan bahwa port C upper dioperasikan sebagai input / output port.
- Bit D2 mendefinisikan mode kerja PPI 8255.
- Bit D1 menunjukkan bahwa port C lower dioperasikan sebagai input / output port.