

# Percobaan

## 1

### JUDUL :

**DASAR INPUT/OUTPUT (1)**  
**(PORT PPI DAN PORT 1 SEBAGAI OUTPUT)**  
Menggunakan DT-51 MinSys

### TUJUAN :

- Mengamati keluaran data berupa nyala LED setelah proses pemindahan data (akses eksternal) dari sebuah register ke alamat memori yang ditunjuk oleh Data Pointer atau I/O port (Port 1 maupun Port PPI).
- Membuat program Output menggunakan PPI Port dan Port 1 pada DT-51 MinSys.

### TEORI :

Pada DT-51 MinSys ada beberapa Port yang bisa digunakan sebagai input/output, yaitu Port 1 mikrokontroler 89C51, serta Port A, Port B, dan Port C dari PPI 82C55.

#### **PPI (Programmable Peripheral Interface)**

Port dari PPI ada 3 yaitu Port A, Port B, dan Port C yang ada pada DT-51 MinSys masing-masing terletak pada alamat 2000h, 2001h, dan 2002h. Sama halnya dengan Port 1, ketiga Port PPI ini juga berjumlah 8 bit dan bisa berfungsi sebagai input maupun output. Hanya saja, kita tidak bisa langsung mengoperasikan Port PPI semudah Port 1. Penggunaan Port PPI harus didahului dengan inialisasi Port melalui *Control Word*-nya yang terletak pada alamat 2003h.

**Control Word** menentukan aktif tidaknya Port PPI dan mode operasi apa yang akan dipakai. Berikut ini merupakan bagan 8 bit Control Word beserta penjelasan singkat.

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

<b>Simbol</b>	<b>Fungsi</b>
D7	Set Flag, bernilai 1 untuk mengaktifkan Port PPI
D6 & D5	Mode Select untuk Port A dan Port C upper (bit 7 – bit 4) bernilai 00 untuk mode 0, 01 untuk mode 1, dan 10 atau 11 untuk mode 2.
D4	Port A bernilai 1 untuk input dan bernilai 0 untuk output.
D3	Port C upper, bernilai 1 untuk input dan bernilai 0 untuk output.
D2	Mode Select untuk Port B dan Port C lower (bit 3 – bit 0) bernilai 0 untuk mode 0, 1 untuk mode 1.
D1	Port B bernilai 1 untuk input dan bernilai 0 untuk output.
D0	Port C lower, bernilai 1 untuk input dan bernilai 0 untuk output.

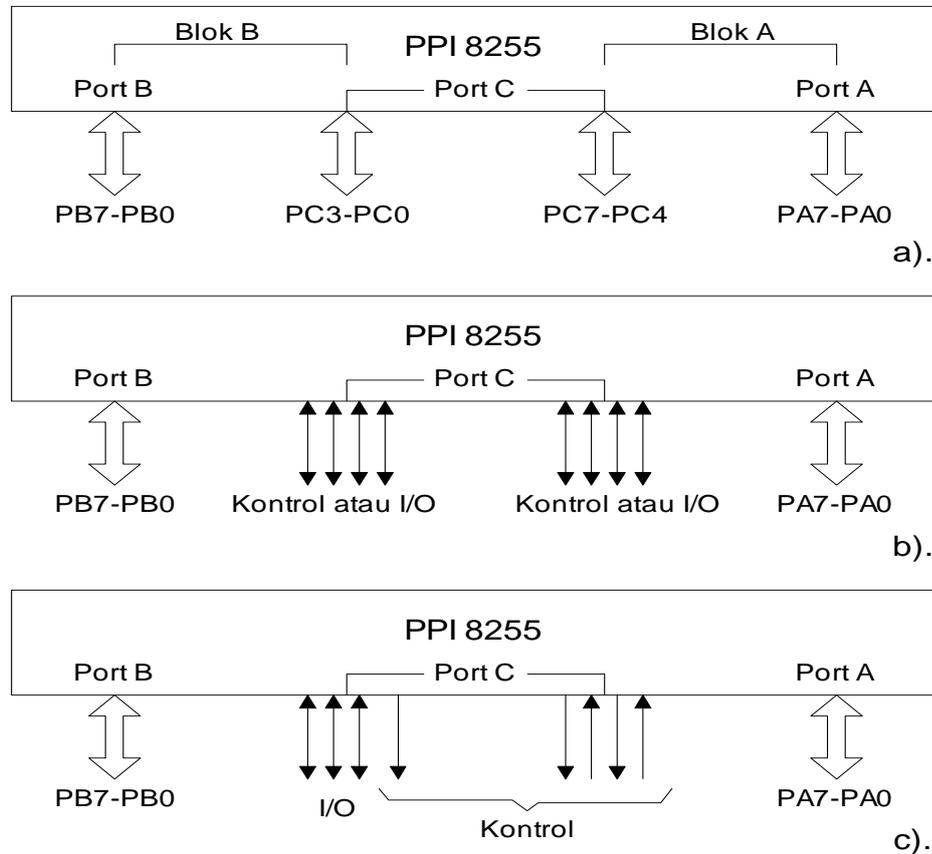
➤ **Mode Operasi PPI 8255**

Ada tiga macam mode operasi yang disediakan PPI 8255. Tiga macam operasi tersebut adalah :

1. Mode 0 adalah mode operasi masukan dan keluaran yang paling sederhana (*basic input/ output operation*). Pada mode ini tiap *port* pada PPI hanya dapat digunakan sebagai masukan atau keluaran.
2. Mode 1 adalah mode operasi masukan dan keluaran yang terkontrol (*strobed input/output operation*). Pada mode ini *port A* dan *port B* dapat digunakan sebagai masukan atau keluaran, tergantung kontrol dari *port C*.
3. Mode 2 adalah mode operasi masukan dan keluaran secara dua arah (*bidirectional input/output operation*). Pada mode ini, *port A* sebagai bus data dua arah yang dikontrol oleh *port C*.

Mode 1 dan 2 adalah mode operasi yang memerlukan sinyal kontrol *input/output* atau *handshaking signal* (sinyal jabat tangan) yaitu sinyal yang menyatakan kesiapan alat

untuk dikenai proses baca atau tulis. Gambaran mode operasi pada PPI 8255 dapat dilihat dalam Gambar 1-1.



Gambar 1-1. Mode Operasi PPI 8255,  
a). mode 0, b), mode 1, c). mode 2.

Dari tiga mode dalam mengoperasikan PPI diatas sebagai dasar dalam praktikum digunakan mode 0, sehingga D6, D5 dan D2 semuanya bernilai 0.

Perlu diketahui bahwa akses Port PPI merupakan akses eksternal. Oleh karena itu ada beberapa perintah dasar dalam pemrograman Port PPI :

- MOV DPTR,<ALAMAT>
- MOV A,<DATA>
- MOVX @DPTR,A
- MOVX A,@DPTR

<ALAMAT> dapat berupa alamat dari Port A, Port B, Port C atau Control Word. <DATA> dapat berupa nilai untuk menginisialisasi Control Word ataupun nilai

yang akan di-output-kan. Sedangkan DPTR adalah Data Pointer yang digunakan untuk mengakses secara eksternal.

➤ **Port PPI Sebagai Keluaran**

Sebagai keluaran, port A, port B, atau port C dapat digunakan sebagai jalur untuk mengeluarkan data, dimana port A dan port B hanya bisa diakses secara *byte* sedangkan port C bisa diakses secara *byte* maupun *bit* dengan cara mengatur format Bit Set/Reset pada control world. Pengertian dari akses per bit adalah mengakses tiap bit secara sendiri – sendiri, tidak secara keseluruhan (8 bit). Sedangkan akses byte adalah mengakses keseluruhan (8 bit) sekaligus. Adapun penjelasan control word untuk bit *set/reset pada Port C* sebagai berikut:

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

<b>Simbol</b>	<b>Fungsi</b>
D7	Set Flag, bernilai 0 untuk mengaktifkan Bit Set/Reset Port C.
D6 – D4	Bit bersifat don't care, disarankan diisi dengan logika '0'.
D3 – D1	Bit Select. Untuk menentukan bit pada port C yang akan diakses. Lebih jelasnya dapat dilihat pada tabel 1-1.
D0	Bit Set/Reset untuk Port C, didisi 1 untuk set dan 0 untuk reset.

Tabel 1-1. Bit Select

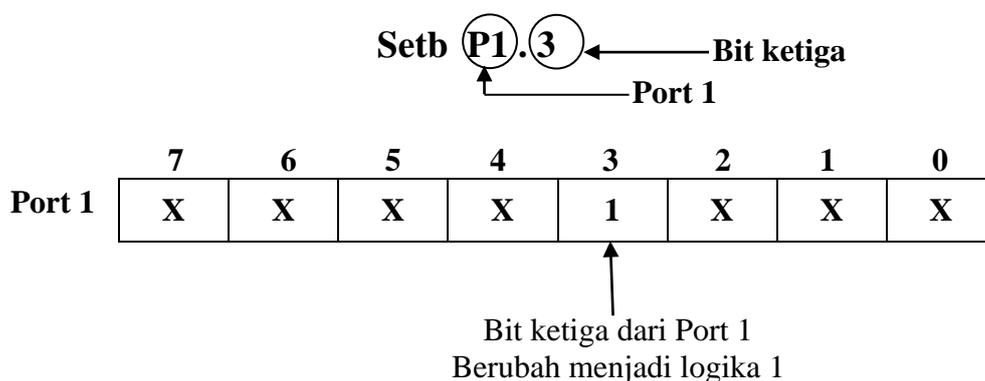
	<b>Bit pada Port C</b>							
	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>D3</b>	1	1	1	1	0	0	0	0
<b>D2</b>	1	1	0	0	1	1	0	0
<b>D1</b>	1	0	1	0	1	0	1	0

**Port 1 89C51**

Pada IC AT89C51 terdapat empat port, yaitu Port 0 , Port 1, Port 2 dan Port 3 yang terletak pada alamat 80H, 90H, A0H dan B0H. Namun, jika digunakan eksternal memori ataupun fungsi-fungsi special, seperti ekxternal interrupt, Serial ataupun External Timer, Port 0, Port 2 dan Port 3 tidak dapat digunakan sebagai port dengan

fungsi umum. Untuk itu disediakan Port 1 yang dikhususkan untuk port dengan fungsi umum. Dalam DT-51 MinSys, Port 0, Port 2 dan Port 3 telah dihubungkan dengan komponen tertentu diantaranya adalah terhubung dengan eksternal memori sehingga tidak bisa digunakan sebagai port dengan fungsi bebas/umum. Sedangkan Port 1 difungsikan sebagai port yang bebas. Port 1 berjumlah 8 bit (Port 1.0 – Port 1.7) dan bisa berfungsi sebagai input maupun sebagai output. Port 1 bisa diakses per byte atau bit sehingga dapat dilakukan perubahan output pada tiap-tiap pin dari port ini tanpa mempengaruhi pin-pin yang lainnya.

*Contoh*, jika dilakukan instruksi Setb P1.3, bit ketiga dari port 1 akan berkeadaan “high” (5V) tanpa mempengaruhi bit-bit yang lain seperti ditunjukkan dalam gambar 1-2.



Gambar 1-2. Bit-bit port

## PERALATAN :

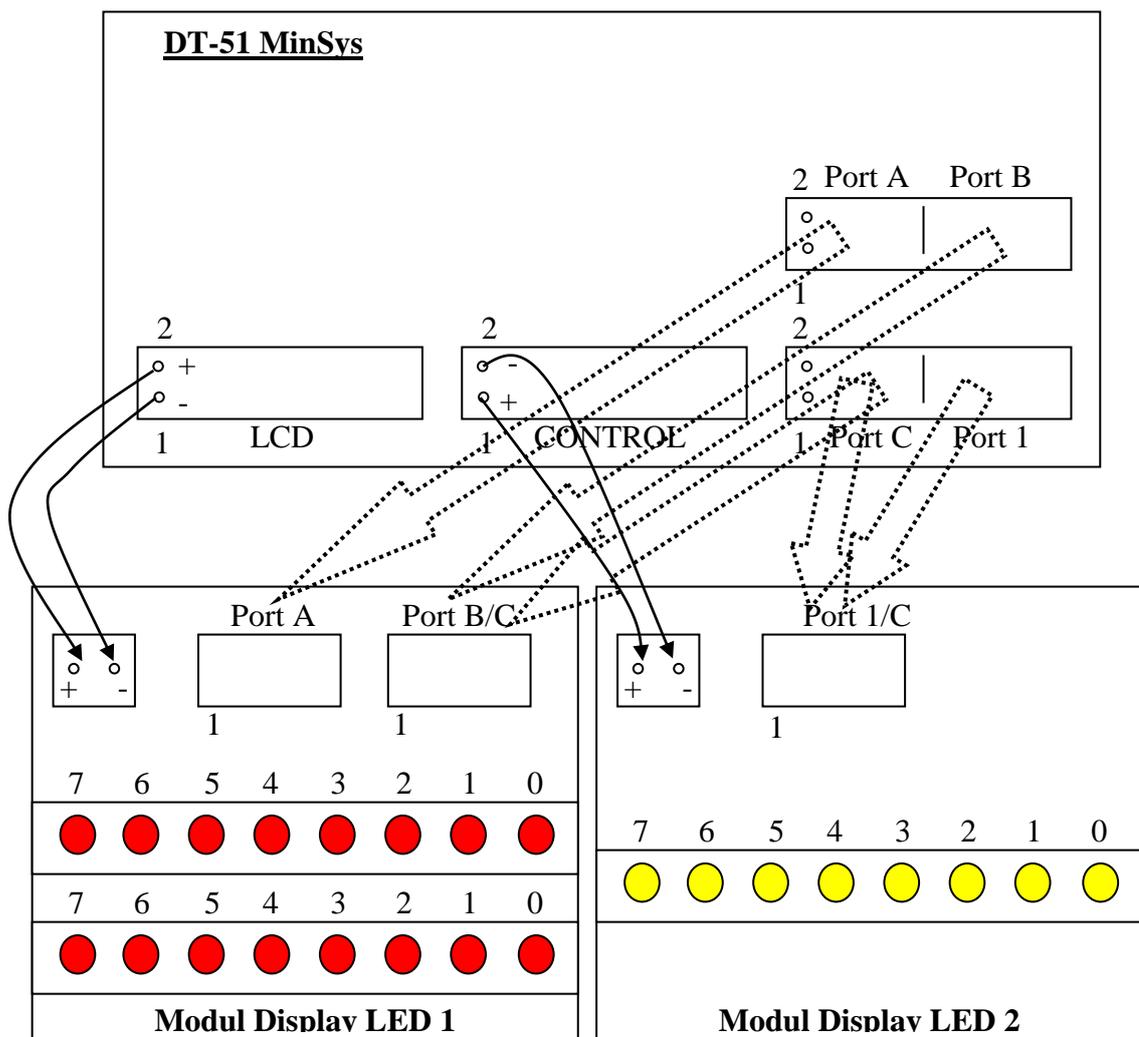
- Modul DT-51
- PC beserta software
  - Debugger
  - Downloader
  - Editor
- LED sebagai piranti output

# PROSEDUR :

## Kombinasi Port A, Port B, Port C dan Port 1 sebagai Output

### 1. Hardware

- Hubungkan konektor (kabel pita) Port 1, Port C pada DT-51 ke *Modul Display LED* jika output yang digunakan adalah Port 1 atau Port C. Tetapi bila outputnya adalah Port A atau Port B maka hubungkan konektor (kabel pita) Port A atau Port B pada DT-51 ke *Modul Display LED* (seperti pada gambar 1-3).



Gambar 1-3. Koneksi DT-51 MinSys dengan Display LED

### Selanjutnya

- Hubungkan kabel serial dari konektor DB-9 pada DT-51 ke PC pada Port serial.

- Hubungkan catu daya 9 Volt AC pada modul DT-51.
- Pin Konektor Select pada posisi Pin 1 dan 2 untuk mode *download program*, pada posisi Pin 2 dan 3 untuk mode *stand alone*, setelah download program berhasil.

## **2. Software**

### ***Editor :***

- Tulislah program pada editor DOS prompt, Notepad atau lainnya.
- Program / file baru tersebut, simpan dalam direktori(folder) DT51 dengan extension *[namafile].ASM*.
- Exit

### ***Debugger :***

- Compile file baru dengan perintah `C:\DT51\ASM51 [nama file].ASM` pada DOS Prompt, dengan terlebih dahulu pindah ke direktori DT51.
- Maka akan terbentuk file-file yang berekstension .HEX, .OBJ, dan .LST disamping itu juga muncul pesan, bahwa file yang telah dicompile tersebut terdapat kesalahan atau tidak.
- Bila ada kesalahan, bisa dilihat letak kesalahannya dengan membuka file yang berekstension .LST, melalui `C:\DT51\edit`.
- Untuk memperbaiki kesalahan (error), buka file yang berekstension .ASM. Jangan lupa disimpan kemudian dicompile kembali.
- Exit

### ***Downloader :***

- Untuk mendownload program yang telah sukses dicompile, maka gunakan perintah `C:\DT51\DT51L [nama file].HEX`
- Apabila proses download selesai, maka akan muncul pesan Download Succeeded.
- Bila gagal, maka akan muncul beberapa prosedur untuk mengecek kembali kesalahan tersebut.

# PROGRAM PERCOBAAN :

## ➤ Akses Port PPI

### - Program 1.1 Test Output data di Port A PPI DT-51

```
$MOD51

                                ORG 4000H ;Awal program DT-51 harus 4000h
;inisialisasi
                                MOV     DPTR,#2003H
                                MOV     A,#80H
                                MOVX    @DPTR,A
;port A
                                MOV     DPTR,#2000H
                                MOV     A,#0CFH
                                MOVX    @DPTR,A
                                SJMP    $
                                END
```

### - Program 1.2 Test Output data di Port B PPI DT-51

```
$MOD51
                                CTRLWRD EQU 2003H ;****deklarasi variabel
                                ORG     4000H
;inisialisasi
                                MOV     DPTR,#CTRLWRD
                                MOV     A,#80H
                                MOVX    @DPTR,A
;port B
                                ULANG:  MOV     DPTR,#2001H
                                MOV     A,#55H
                                MOVX    @DPTR,A
                                JMP     ULANG
                                END
```

### - Program 1.3 Test Output data di Port C PPI DT-51

```
$MOD51
                                CSEG
                                ORG     4000H
                                JMP     START

                                ORG 4100H
START:  MOV     DPTR,#2003H ;inisialisasi
                                MOV     A,#80H
                                MOVX    @DPTR,A

                                MOV     DPTR,#2002H ;port C
                                MOV     A,#0AAH
                                MOVX    @DPTR,A
                                SJMP    $
                                END
```

**- Program 1.4 Test Output data di Port A dab Port B PPI**

```

$MOD51

PORTA      EQU    2000H ; *****
PORTB      EQU    2001H ; deklarasi variabel
PORTC      EQU    2002H ;
PORTCW     EQU    2003H ; *****
CW         EQU    0080H ; ***semua jadi output
          ORG    4000H
          JMP    START

          ORG    4100H
;inisialisasi
START:     MOV     DPTR,#PORTCW
          MOV     A,#CW
          MOVX    @DPTR,A

          ULANG:  MOV     DPTR,#PORTA
          MOV     A,#0AAH
          MOVX    @DPTR,A
          MOV     DPTR,#PORTB
          MOV     A,#55H
          MOVX    @DPTR,A
          JMP     ULANG
          END

```

**- Program 1.5 Pembuatan lampu kedip degan delay di Port A**

```

$MOD51

PORTA      EQU    2000H ; *****
PORTB      EQU    2001H ; deklarasi variabel
PORTC      EQU    2002H ;
PORTCW     EQU    2003H ; *****
CW         EQU    0080H ; *semua jadi output

          ORG    4000H
          AJMP   START
          ORG    4100H

          DELAY:  MOV     R0,#03H
          DEL1A:  MOV     R1,#0FFH
          DEL1B:  MOV     R2,#0FFH
          DJNZ   R2,$
          DJNZ   R1,DEL1B
          DJNZ   R0,DEL1A
          RET

;inisialisasi
START:     MOV     DPTR,#PORTCW
          MOV     A,#CW
          MOVX    @DPTR,A

;buat lampu berkedip
LOOP:     MOV     A,#0FFH
          MOV     DPTR,#PORTA
          MOVX    @DPTR,A
          ACALL   DELAY

```

```

MOV      A, #00H
MOVX    @DPTR, A
ACALL   DELAY
AJMP    LOOP
END

```

➤ **Akses Port 1**

- ***Program 1.6 Test Output data di Port 1 DT-51 (akses per bit)***

\$MOD51

```

CSEG
ORG 4000H
JMP START
ORG 4100H
START: MOV SP, #30H
      SETB P1.0
      CLR P1.1
      SETB P1.2
      CLR P1.3
      SETB P1.4
      SETB P1.5
      SETB P1.6
      SETB P1.7
      SJMP $
      END

```

- ***Program 1.7 Test Output data di Port 1 DT-51 (akses byte)***

\$MOD51

```

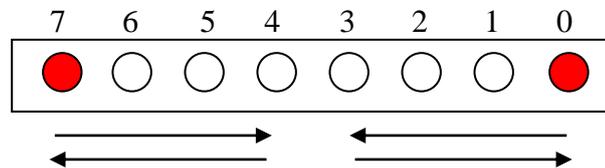
ORG 4000H
START: MOV A, #01H
LAJU:  MOV P1, A
      ACALL DELAY
      INC A
      CJNE A, #00H, LAJU
      JMP START

DELAY: MOV R7, #03H
DEL1:  MOV R6, #0FFH
DEL2:  MOV R5, #0FFH
      DJNZ R5, $
      DJNZ R6, DEL2
      DJNZ R7, DEL1
      RET
      END

```

## TUGAS :

1. Dengan menggunakan instruksi logika dan rotasi, buatlah program untuk menyalakan LED di Port C dengan tampilan sebagai berikut :



2. Buatlah program untuk menyalakan LED 0 sampai LED 7 melompat 2 menggunakan instruksi rotasi, nyala LED dimulai dari LED 0, kemudian LED 2, 4, 6 kembali ke LED 0 berulang terus menerus.
3. Buat program untuk menyalakan satu LED di Port A ke Port B, nyala LED dimulai dari LED 0 pada port A seperti arah panah yang ditunjukkan pada gambar dibawah :

