



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

BAGIAN 1

Memprogram Port sebagai Output dan Input Sederhana

Tujuan Pembelajaran Umum:

1. Mahasiswa trampil memprogram Port sebagai Input dan Output sederhana menggunakan bahasa pemrograman assembly

Tujuan Pembelajaran Khusus:

1. Mahasiswa memahami Konstruksi Port
2. Mahasiswa memahami pengolahan program akses input sesuai kebutuhan pengambilan data
3. Mahasiswa memahami pengolahan program akses output sesuai kebutuhan pengiriman data

Konstruksi Port Paralel AT89S51

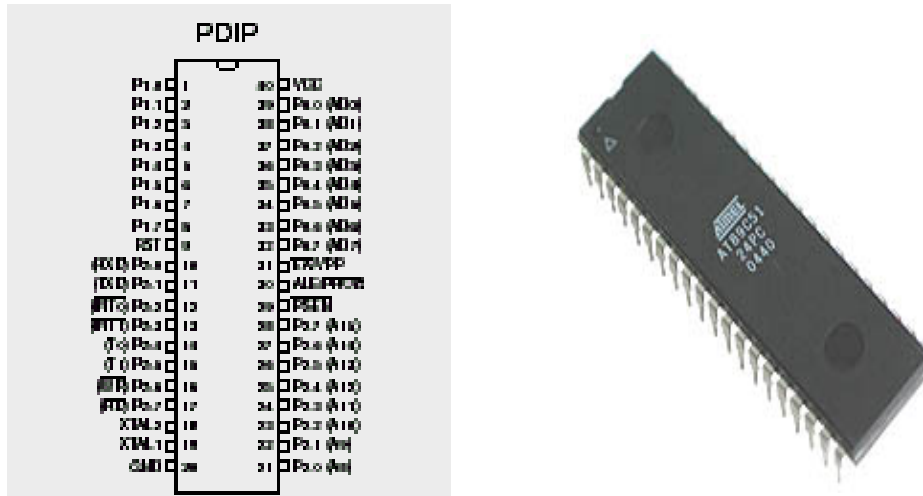
Port paralel dan port serial merupakan sarana utama sebuah chip Mikrokontroler. Lewat sarana ini mikrokontroler mengeluarkan dan menerima sinyal digital yang dipakai mengendalikan rangkaian-rangkaian di luar chip secara langsung. Biasanya port paralel berfungsi ganda, jika dikehendaki port paralel bisa diatur dipakai untuk keperluan lain seperti untuk saluran data atau saluran alamat.

Chip mikrokontroler AT89S51 mempunyai 40 pin, 32 pin di antaranya adalah pin untuk keperluan port paralel. Satu port paralel terdiri dari 8 pin, dengan demikian 32 pin tersebut membentuk 4 buah port paralel, yang masing-masing dikenali sebagai Port 0, Port 1, Port 2 dan Port 3. Masing-masing jalur (pin) dari port paralel dinomori mulai dari 0 sampai 7, jalur (pin) pertama Port 0 disebut sebagai **P0.0**, jalur terakhir Port 3 adalah **P3.7**.



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana



Gambar 23. Bentuk Fisik dan Susunan Pin AT89S51

Keluarga AT89S51 mempunyai dua kelompok instruksi untuk mengeluarkan data ke port paralel:

- Kelompok instruksi pertama bekerja pada port seutuhnya artinya 8 jalur dari port bersangkutan, misalnya **MOV P1, #0FF** membuat ke-delapan jalur port 1 sekaligus menjadi '1', **MOV P1, #00** membuat ke-delapan jalur port 1 sekaligus menjadi '0'. Pola semacam ini disebut dengan pola pengalamatan byte.
- Kelompok instruksi kedua hanya berpengaruh pada salah satu jalur dari port, misalnya **SETB P3.4** membuat **P3.4** menjadi '1' atau **CLR P3.3** membuat **P3.3** menjadi '0'. Pola pengalamatan semacam ini disebut pola pengalamatan bit.

Selain itu port paralel bisa pula dipakai untuk menerima sinyal digital dari rangkaian di luar chip mikrokontroler:

- Instruksi **MOV A, P1** mengambil sinyal digital pada semua jalur Port 1 dan menyimpannya di Akumulator.
- Pengambilan sinyal bisa hanya pada satu jalur port, instruksi **JNB P3.7,\$** dipakai untuk memantau jalur **P3.7**, jika **P3.7 = '0'** AT89S51 akan melompat ke alamat relative yang ditentukan, AT89S51 baru akan meneruskan instruksi yang ada di bawah instruksi tersebut jika **P3.7='1'**. Instruksi yang berlawanan dengan **JNB** adalah **JB**.



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Satu hal yang perlu diperhatikan, agar sinyal dari luar bisa dibaca dengan benar, sebelum membaca jalur port bersangkutan harus terlebih dulu dibuat menjadi '1'. Hal ini akan dijelaskan lebih lanjut di bagian bawah.

Port 0, Port 2 dan Port 3 bisa berfungsi ganda, selain dipakai sebagai port paralel bisa dipakai untuk keperluan lain.

- Kegunaan lain dari Port 0 adalah sebagai saluran-data (*data bus* - **D0..D7**) dan setengah saluran-alamat (*address bus* - **A0..A7**) yang dipakai untuk memasang memori tambahan di luar chip mikrokontroler.
- Kapasitas memori tambahan bisa sampai 64 KiloByte sehingga memerlukan 16 jalur saluran alamat, jalur alamat **A0..A7** dilewatkan Port 0 di atas, sedangkan setengah saluran alamat yang lain (**A8,,A15**) dilewatkan Port 2.
- Selain untuk port paralel, Port 3 bisa pula dipakai untuk berbagai macam keperluan sarana input/output dengan keperluan khusus.

Untuk menunjang semua kemampuan yang disebutkan di atas, rangkaian elektronik pembentuk port mempunyai konstruksi khusus, bahkan konstruksi masing-masing port berlainan. Berikut adalah pembahasan tentang konstruksi rangkaian elektronik masing-masing port.

Konstruksi Port 1

Port 1 merupakan satu-satunya port AT89S51 yang tidak berfungsi ganda, sehingga konstruksi Port 1 merupakan dasar konstruksi bagi port-port lain. Port yang lain dibangun atas dasar Port 1 dengan beberapa modifikasi.

Gambar 24 memperlihatkan konstruksi salah satu jalur dari Port 1, inti dari rangkaian ini adalah sebuah *latch*, *output Q** dari *latch* dihubungkan ke rangkaian *inverter* yang dibentuk dengan transistor MOS dan tahanan yang dihubungkan ke Vcc, output inverter dihubungkan ke pin chip AT89S51. Input *latch* terhubung ke salah satu jalur saluran-data di dalam chip AT89S51, saat intruksi semacam **MOV P1,A** dijalankan, pada input clock dari *latch* akan timbul pulsa yang membuat isi jalur data direkam ke dalam *latch*.

Data '1' yang direkam ke dalam *latch* membuat *Q** menjadi '0' sehingga transistor MOS *off*, tegangan pada pin chip menjadi Vcc yang dikenali sebagai '1'. Sebaliknya data '0' dalam *latch* membuat *Q** menjadi '1' sehingga transistor MOS *on*, menghubungkan pin chip ke *ground*, hal ini dikenali sebagai '0'.



MODUL 5

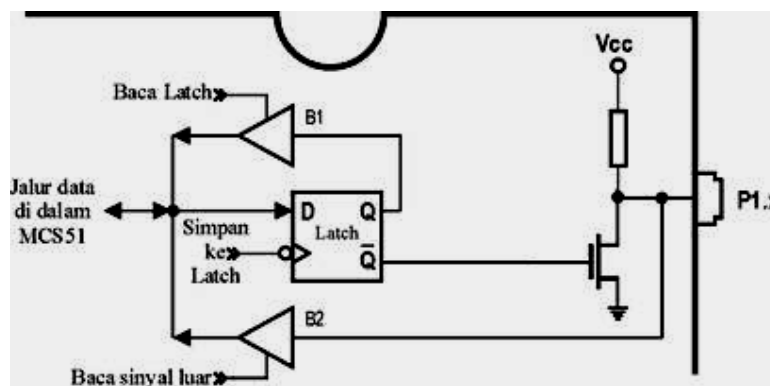
Memprogram Port sebagai Output dan Input Sederhana

Di luar rangkaian *latch* dan *inverter*, rangkaian port 1 masih dilengkapi pula dengan penyangga 3 kondisi (*tri-state buffer*) B1 dan B2 yang dipakai sebagai jalur input ke jalur saluran-data di dalam chip AT89S51.

Saat AT89S51 melaksanakan instruksi-instruksi yang bersifat ambil-rubah-simpan (*read-modify-write*), misalnya instruksi **INC P1**; **DEC P1**; **ANL P1,A**; **ORL P1,A**; **XRL P1,A**, output *Q* dari *latch* diambil lewat penyangga *B1*, diolah di dalam inti prosesor AT89S51 kemudian dikembalikan ke *latch*.

B2 dipakai saat AT89S51 melaksanakan perintah **MOV A,P1** atau **JB P1.x,...** atau **JNB P1.x,...** menghubungkan pin P1.x ke jalur data dalam chip. Tapi hasil pembacaan ini baru bisa terjadi dengan baik kalau transistor MOS dalam keadaan *off*, sehingga tegangan yang ada di pin chip bisa sampai ke jalur data dalam chip AT89S51 tanpa gangguan. Sebaliknya kalau transistor MOS dalam keadaan *on* maka tegangan pada pin chip akan disalurkan ke ground, sehingga data yang sampai ke jalur data dalam chip AT89S51 selalu bernilai '0'. Meng-*off*-kan transistor MOS bisa dilakukan dengan cara menjalankan instruksi **MOV P1,#0FF** atau **SETB P1.x**.

Di dalam chip AT89S51 terdapat 8 buah rangkaian semacam Gambar 14 yang membentuk Port 1.



Gambar 24
Konstruksi Port 1

Konstruksi Port 3

Selain sebagai port paralel, Port 3 bisa pula dipakai untuk berbagai keperluan bermacam sarana input/output dan fasilitas untuk menambah memori data di luar chip AT89S51. Fungsi lain dari masing-masing pin Port 3 antara lain adalah **P3.0** berfungsi pula sebagai **RxD** yakni input data seri dari UART, **P3.1** berfungsi pula sebagai **TxD** yakni output data seri dari UART, **P3.2** dan **P3.3**

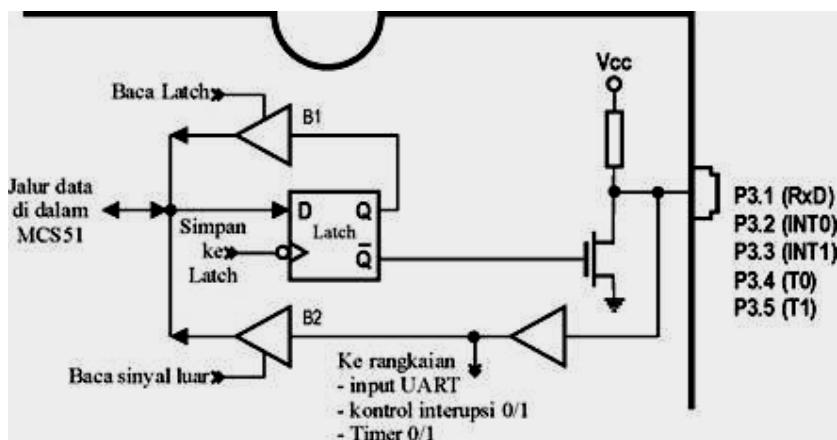


MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

masing-masing bisa berfungsi sebagai **INT0*** dan **INT1***, yang bertugas menerima sinyal permintaan interupsi dari rangkaian di luar chip AT89S51. **P3.4** dan **P3.5** masing-masing bisa berfungsi sebagai **T0** dan **T1**, yang bertugas menerima sinyal denyut dari rangkaian di luar chip AT89S51, saat Timer 0 atau Timer 1 dikerjakan sebagai 'counter'. **P3.6** dan **P3.7** bisa menjadi **WR*** dan **RD***, yang dipakai untuk mengendalikan memori data yang ditambahkan di luar chip AT89S51.

Konstruksi Port 3 dibagi menjadi 2 model, model pertama pada **P3.0**, **P3.6** dan **P3.7** yang dipakai untuk menyalurkan sinyal keluar dari chip AT89S51, seperti terlihat dalam Gambar 25a. Transistor MOS dikendalikan bersama oleh output *Q* dari *Latch* dan rangkaian output yang ada di dalam chip AT89S51 lewat sebuah gerbang *NAND*. Dari rangkaian tersebut jelas terlihat agar output fungsi alternatif bisa diteruskan ke masing-masing pin dengan benar, sebelumnya *latch* harus diisi dengan '1'.



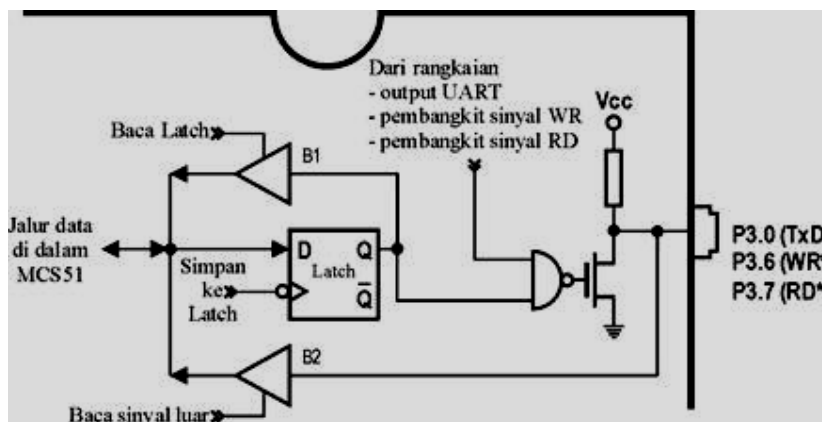
Gambar 25a
Konstruksi Port 3 untuk sarana output khusus

Model konstruksi yang kedua ada pada Port 3.1, Port 3.2, Port 3.3, Port 3.4 Port 3.5, rangkaian ini mirip sekali dengan rangkaian Port 1 hanya saja sinyal dari pin chip AT89S51 selain disalurkan ke jalur data di dalam AT89S51, disalurkan juga ke rangkaian input di dalam chip AT89S51.



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana



Gambar 25b
Konstruksi Port 3 untuk sarana input khusus

Konstruksi Port 2

Port 2 bisa juga dipakai sebagai saluran alamat, untuk keperluan ini rangkaian dasar Port 1 ditambah dengan sebuah gerbang *AND*, sebuah transistor MOS dan sebuah inverter. Gambar 26 menggambarkan Port 2 sedang berfungsi sebagai Port Paralel, transistor MOS *T2* dalam keadaan '*off*', keadaan saat ini sama persis dengan keadaan Port 1.

Pada dasarnya Port 2 masih tetap berfungsi sebagai port paralel, hanya pada saat diperlukan saja Port 2 berubah fungsi sebagai saluran alamat dan sesudah itu akan segera kembali berfungsi sebagai port paralel. Saat Port 2 berfungsi sebagai saluran alamat input A gerbang *AND* menjadi '1', jalur alamat di dalam chip AT89S51 lewat gerbang *AND* mengendalikan transistor MOS *T2*, saklar *S* berpindah posisi ke kanan menghubungkan jalur alamat yang sudah dibalik inverter ke rangkaian transistor MOS *T1*.

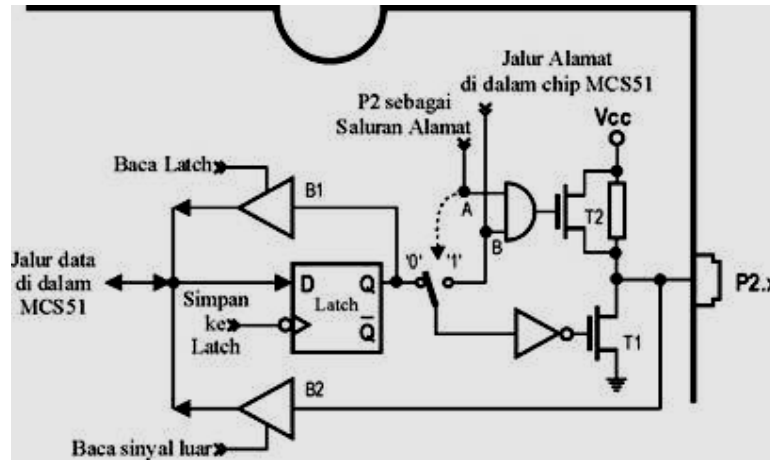
Dengan adanya inverter pada rangkaian Gate *T1*, mengakibatkan keadaan transistor MOS *T1* dan *T2* selalu berbalikan. Saat jalur alamat di dalam chip bernilai '1' transistor MOS *T2* '*on*' menghubungkan pin *P2.x* ke *Vcc* dan transistor MOS *T1* '*off*', sehingga *P2.x* berada dalam keadaan '1'. Sebaliknya, jalur alamat di dalam chip bernilai '0' transistor MOS *T2* '*off*' dan transistor MOS *T1* '*on*' menghubungkan kakami *P2.x* ke ground, sehingga *P2.x* berada dalam keadaan '1'.

Saat Port 2 dipakai sebagai saluran alamat, isi semua latch pada Port 2 tidak berubah, sehingga meskipun Port 2 masih bisa berfungsi sebagai port output paralel dengan baik.



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana



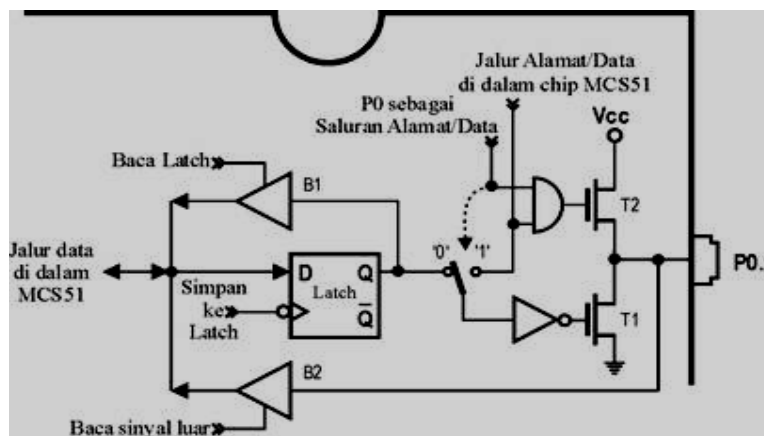
Gambar 26 Konstruksi Port 2

Konstruksi Port 0

Konstruksi Port 0 identik dengan konstruksi Port 2, bedanya tahanan di atas transistor MOS T1 dihilangkan. Saat Port 0 dipakai sebagai saluran alamat/data, mekanisme kerjanya sama dengan Port 2.

Port 0 tidak mempunyai tahanan yang terhubung ke Vcc yang dipasang di atas transistor MOS T1, seperti pada konstruksi port-port yang lain. Pada saat Port 0 dipakai sebagai port output tegangan pada pin P0.x tidak mungkin menjadi '1', untuk mengatasi hal ini, kalau memakai Port 0 sebagai port output harus dipasang tahanan ke Vcc di luar chip AT89S51.

Setiap kali Port 0 dipakai untuk saluran alamat/data, semua latch pada Port 0 dibuat menjadi '1', sehingga kalau Port 0 dipakai untuk saluran alamat/data maka Port 0 sudah tidak dapat berfungsi sebagai port paralel lagi.



Gambar 27
Konstruksi Port 0



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

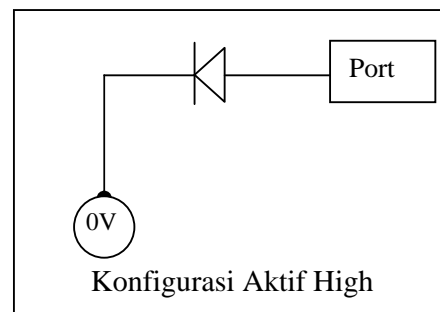
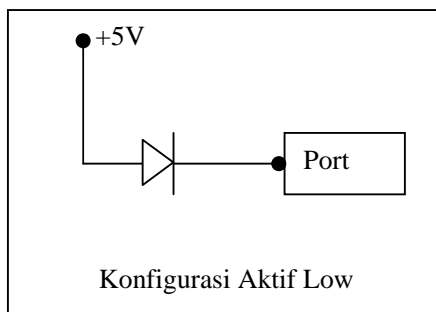
Program Akses I/O

Untuk dapat mengakses Port sebagai output sederhana dapat dipelajari menggunakan indikator LED. LED adalah dioda emisi cahaya yang akan menyala jika mendapat bias maju. Menyala atau padam sebuah LED tergantung bias pemberian logika. Pada tabel 9 ditunjukkan tabel kebenaran penyalan LED.

Tabel 9. Tabel kebenaran penyalan LED.

No	ANODA	KATODA	KEADAAN LED
1.	0	0	Padam
2.	0	1	Padam
3.	1	0	Menyala
4.	1	1	Padam

Dari tabel 9 terlihat hanya ada satu keadaan untuk menyalakan LED, dan tiga keadaan untuk memadamkan LED. Berdasarkan tabel 9 dapat dibuat konfigurasi LED aktif LOW dimana Anoda dapat catu + 5 Volt dan Katoda dihubungkan dengan Port. Sedangkan untuk konfigurasi aktif HIGH, Katoda dihubungkan ke ground dan Anoda dihubungkan ke Port.



Pada konfigurasi aktif Low, LED akan menyala jika Port berkeadaan logika "0" dan LED akan padam jika Port berkeadaan logika "1". Sedangkan pada konfigurasi aktif high akan terjadi sebaliknya LED akan menyala jika Port berkeadaan logika "1" dan LED akan padam jika Port berkeadaan logika "0". Konfigurasi ini menjadi sangat penting untuk diperhatikan dalam kaitannya dengan pembentukan data yang akan di kirim ke Port sebagai data output.

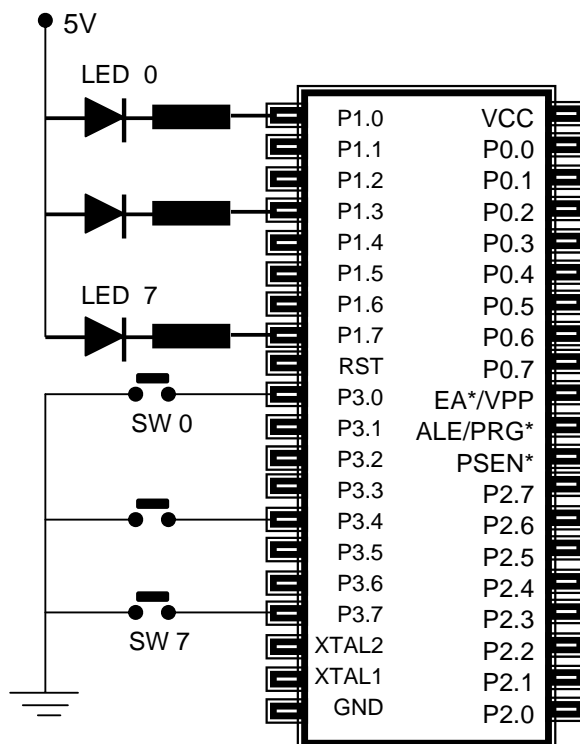


MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Memprogram Port Output Input

Gambar 28 di bawah ini dapat digunakan sebagai kasus pengembangan program untuk akses port sebagai out put. Ada delapan buah LED (L0, L1, L2, L3, L4, L5, L6, dan L7) katodanya dihubungkan ke Port 1 dan anodanya diberi tegangan 5 volt. Konfigurasi rangkaian gambar 28 jelas merupakan konfigurasi aktif Low.



Gambar 28. Konfigurasi Port I/O

Untuk menyalakan LED port diberi logika "0" dan untuk memadamkan port diberi logika "1". Sebagai suatu contoh jika LED0 sampai dengan LED3 padam dan LED4 sampai dengan LED7 menyala maka format datanya adalah sebagai berikut :

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0	DATA
0	0	0	0	1	1	1	1	0FH

Menggunakan data "00001111B" atau "0FH" dapat dibuat program sebagai berikut :



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

PROGRAM MODUL 5-1

```
-----  
;Program Penyalaaan LED pada Port 1  
;LED 0-3 Padam ; Led 4 - 7 Nyala  
;Nama File Modul51.asm  
-----  
ORG 0H ; program dimulai dari alamat 0000 pada Plash Memori  
Mulai: MOV P1,#00FH ; Kirim data 0FH untuk menyalakan LED4-LED7  
SJMP Mulai ; Kembali ke Label mulai agar LED tetap bekerja  
END ; Akhir program
```

Pada baris pertama `ORG 0H` menunjuk penempatan program pada alamat awal 0000H. Perintah `MOV P1,#0FH` bekerja mengeluarkan data 0FH ke port 1 untuk membentuk penyalaaan LED4 – LED7. Perintah `SJMP Mulai` bekerja membangun loop sehingga program counter berputar terus menerus.

Untuk membangun port sebagai fungsi input dalam pengambilan data dari luar mikrokontroler dapat menggunakan model konfigurasi gambar 28. Delapan buah Push Button dipasang pada Port 3 dengan konfigurasi aktif high. Dalam keadaan tidak ditekan input memberikan data “1” dan jika ditekan memberikan data “0”. Program MODUL 5-2 dapat digunakan sebagai contoh pemanfaatan port untuk pengambilan data.

PROGRAM MODUL 5-2

```
-----  
; Program Baca Input Port 3  
; Disajikan dengan Penyalaaan LED pada Port 1  
; Tombol ditekan LED menyala  
; Nama File Modul52.asm  
-----  
ORG 0h  
Mulai:  
MOV A,P3 ; baca tombol pada Port 3 dan catat di Reg A  
MOV P1,A ; keluarkan data pada Reg A ke Port 1  
SJMP Mulai ; lompat ke label mulai  
END
```



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

BAGIAN 2

PETUNJUK KERJA

A. PETUNJUK PRE-TEST

1. Kerjakan soal pre-test yang ada pada Modul 5 dengan mengisi tanda cek.
2. Isi dengan sebenarnya sesuai keadaan saudara
3. Jika saudara telah memiliki kompetensi seperti yang dinyatakan dalam pre test kerjakan soal-soal Post-Test
4. Jika saudara belum memiliki kompetensi seperti yang dinyatakan dalam pre test pelajari materi pada bagian satu dari Modul ini

B. PETUNJUK POST-TEST

I. UMUM

Dalam tugas ini, pada akhirnya saudara akan memiliki kompetensi terkait dengan :

1. Memahami konstruksi port
2. Memahami pengolahan program akses input sesuai kebutuhan pengambilan data.
3. Memahami pengolahan program akses output sesuai kebutuhan pengiriman data

II. KHUSUS

1. Kerjakan kasus-kasus program pada bagian post test sampai pada pengujian hasilnya pada down loader atau in system programming.



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

BAGIAN 3 PRE-TEST

Subkompetensi	Pernyataan	Saya memiliki kompetensi ini	
		Tidak	Ya
5. Memprogram Port sebagai Output dan Input Sederhana	5.1. Saudara memahami Konstruksi Port mikrokontroler AT89S51		
	5.2. Saudara memahami pengolahan program akses input dengan baik sesuai kebutuhan pengambilan data		
	5.3. Saudara memahami pengolahan program akses output dengan baik sesuai kebutuhan pengiriman data		



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

BAGIAN 4

POST-TEST

1. Buatlah program yang bekerja menyalakan satu buah lampu LED bergeser mulai dari LED0 menuju lampu LED7.
2. Buatlah program yang bekerja menyalakan dua buah lampu LED bergeser mulai dari LED1 dan LED0 menuju lampu LED7
3. Buatlah program yang bekerja menyalakan satu buah lampu LED bergeser mulai dari LED7 menuju lampu LED0.
4. Buatlah program yang bekerja menyalakan dua buah lampu LED bergeser mulai dari LED7 dan LED6 menuju lampu LED1
5. Buatlah program yang bekerja menyalakan dua buah lampu LED bergeser mulai dari tengah LED4 dan LED3 menuju lampu LED5 dan LED2 terus ke LED6 dan LED1 berhenti di LED7 dan LED0.
6. Buat program yang bekerja menyalakan seluruh lampu LED berkedip hidup-mati
7. Buat program jika tombol input SW0 ditekan LED berkedip hidup-mati dan tidak ada tombol ditekan LED menyala semua.



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

BAGIAN 5

KUNCI JAWABAN

Soal no 1.

PROGRAM MODUL 5-3

```
-----  
; Program Lampu nyala bergeser kekiri mulai dari LED0 di Port 1  
; Nama File Modul53.asm  
-----  
ORG      0h  
MOV      A,#1111110B    ; Masukkan data 1111110B ke Accumulator  
Mulai:   MOV      P1,A      ; keluarkan isi Accumulator ke Port 1  
         ACALL   Delay     ; panggil sub routine Delay  
         RL      A         ; Rotate Accumulator left  
         :      <- b7- b6- b5- b4- b3 - b2 - b1- b0 <-----  
         ;      |----->-----|  
         SJMP   Mulai     ; Lompat ke alamat dg label Mulai  
  
-----  
; sub routine delay  
-----  
Delay:   MOV      R0,#0FFh    ; Isi Register R0 dengan FF  
Delay1:  MOV      R1,#0FFh    ; Isi Register R1 dengan FF  
Delay2:  DJNZ     R1,Delay2    ; Kurangi R1 dengan 1, bila hasil belum  
         ; sama dengan 0 maka lompat ke Delay2  
         DJNZ     R0,Delay1    ; Kurangi R0 dengan 1, bila hasil belum  
         ; sama dengan 0 maka lompat ke Delay1  
         RET      ; Kembali ke alamat setelah perintah  
         ; 'Acall Delay'  
End
```



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Soal no 2.

PROGRAM MODUL 5-4

```
-----  
; Program Lampu nyala bergeser kekiri mulai dari LED0 di Port 1  
; Nama File Modul54.asm  
-----  
ORG      0h  
MOV      A,#11111100B    ; Masukkan data 11111100B ke Accumulator  
Mulai:   MOV      P1,A      ; keluarkan isi Accumulator ke Port 1  
         ACALL   Delay     ; panggil sub routine Delay  
         RL      A         ; Rotate Accumulator left  
         ; ←- b7- b6- b5- b4- b3 - b2 - b1- b0 ←-----  
         ; |-----→-----|  
         SJMP   Mulai     ; Lompat ke alamat dg label Mulai  
  
-----  
; sub routine delay  
-----  
Delay:   MOV      R0,#0FFh    ; Isi Register R0 dengan FF  
Delay1:  MOV      R1,#0FFh    ; Isi Register R1 dengan FF  
Delay2:  DJNZ     R1,Delay2    ; Kurangi R1 dengan 1, bila hasil belum  
         ; sama dengan 0 maka lompat ke Delay2  
         DJNZ     R0,Delay1    ; Kurangi R0 dengan 1, bila hasil belum  
         ; sama dengan 0 maka lompat ke Delay1  
         RET      ; Kembali ke alamat setelah perintah  
         ; 'Acall Delay'  
End
```



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Soal no 3.

PROGRAM MODUL 5-5

```
-----  
; Program Lampu nyala bergeser kekanan mulai dari LED7 di Port 1  
; Nama File Modul55.asm  
-----  
ORG      0h  
MOV      A,#01111111B    ; Masukkan data 01111111B ke Accumulator  
Mulai:   MOV      P1,A      ; keluarkan isi Accumulator ke Port 1  
         ACALL   Delay     ; panggil sub routine Delay  
         RR      A         ; Rotate Accumulator left  
         ;      :  →- b7- b6- b5- b4- b3 - b2 - b1- b0 →-----  
         ;      : |-----←-----|  
         SJMP   Mulai     ; Lompat ke alamat dg label Mulai  
  
-----  
; sub routine delay  
-----  
Delay:   MOV      R0,#0FFh    ; Isi Register R0 dengan FF  
Delay1:  MOV      R1,#0FFh    ; Isi Register R1 dengan FF  
Delay2:  DJNZ     R1,Delay2    ; Kurangi R1 dengan 1, bila hasil belum  
         ; sama dengan 0 maka lompat ke Delay2  
         DJNZ     R0,Delay1    ; Kurangi R0 dengan 1, bila hasil belum  
         ; sama dengan 0 maka lompat ke Delay1  
         RET      ; Kembali ke alamat setelah perintah  
         ; 'Acall Delay'  
End
```




MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Soal no 4.

PROGRAM MODUL 5-6

```
-----  
; Program dua Lampu nyala bergeser kekanan mulai dari LED7 dan LED6 di Port 1  
; Nama File Modul56.asm  
-----  
ORG 0h  
MOV A,#00111111B ; Masukkan data 00111111B ke Accumulator  
Mulai: MOV P1,A ; keluarkan isi Accumulator ke Port 1  
ACALL Delay ; panggil sub routine Delay  
RR A ; Rotate Accumulator left  
; →- b7- b6- b5- b4- b3 - b2 - b1- b0 →-----  
; |-----←-----|  
SJMP Mulai ; Lompat ke alamat dg label Mulai  
  
-----  
; sub routine delay  
-----  
Delay: MOV R0,#0FFh ; Isi Register R0 dengan FF  
Delay1: MOV R1,#0FFh ; Isi Register R1 dengan FF  
Delay2: DJNZ R1,Delay2 ; Kurangi R1 dengan 1, bila hasil belum  
; sama dengan 0 maka lompat ke Delay2  
DJNZ R0,Delay1 ; Kurangi R0 dengan 1, bila hasil belum  
; sama dengan 0 maka lompat ke Delay1  
RET ; Kembali ke alamat setelah perintah  
; 'Acall Delay'  
End
```



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Soal no 5.

PROGRAM MODUL 5-7

```
;-----  
; Program 2 Lampu nyala bergeser dari tengah mulai dari LED4 dan LED3 di Port 1  
; Nama File Modul57.asm  
;-----
```

```
ORG 0H  
Mulai: MOV DPTR,#ANGKA  
MOV R5,#05H  
Putar: CLR A  
MOVC A,@A+DPTR  
MOV P1,A  
INC DPTR  
LCALL Delay  
DJNZ R5,Putar  
SJMP Mulai
```

```
ANGKA: DB 0FFh,0E7H,0DBH,0BDH,07EH
```

```
;-----  
; sub routine delay  
;-----  
Delay: MOV R0,#0FFh ; Isi Register R0 dengan FF  
Delay1: MOV R1,#0FFh ; Isi Register R1 dengan FF  
Delay2: DJNZ R1,Delay2 ; Kurangi R1 dengan 1, bila hasil belum  
; sama dengan 0 maka lompat ke Delay2  
DJNZ R0,Delay1 ; Kurangi R0 dengan 1, bila hasil belum  
; sama dengan 0 maka lompat ke Delay1  
RET ; Kembali ke alamat setelah perintah  
; 'Acall Delay'  
End
```



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Soal no 6.

PROGRAM MODUL 5-8

```
-----  
; Program Lampu nyala LED berkedip  
; Nama File Modul58.asm  
-----  
      ORG 0H  
Ulang : MOV  P1,#00H  
        LCALL Delay  
        MOV  P1,#0FFH  
        LCALL Delay  
        SJMP Ulang  
  
-----  
; sub routine delay  
-----  
Delay: MOV   R0,#0FFh      ; Isi Register R0 dengan FF  
Delay1: MOV   R1,#0FFh      ; Isi Register R1 dengan FF  
Delay2: DJNZ  R1,Delay2     ; Kurangi R1 dengan 1, bila hasil belum  
                               ; sama dengan 0 maka lompat ke Delay2  
        DJNZ  R0,Delay1     ; Kurangi R0 dengan 1, bila hasil belum  
                               ; sama dengan 0 maka lompat ke Delay1  
        RET                ; Kembali ke alamat setelah perintah  
                               ; 'Acall Delay'  
End
```



MODUL 5

Memprogram Port sebagai Output dan Input Sederhana

Soal no 7.

PROGRAM MODUL 5-9

```
-----  
; Program Lampu nyala LED berkedip  
; Nama File Modul59.asm  
-----  
      ORG    0H  
Ulang :  
      MOV    P1,#00H  
      JB     P3.0,Ulang  
  
      MOV    P1,#00H  
      LCALL Delay  
      MOV    P1,#0FFH  
      LCALL Delay  
      SJMP   Ulang  
  
-----  
; sub routine delay  
-----  
Delay:  MOV    R0,#0FFh      ; Isi Register R0 dengan FF  
Delay1: MOV    R1,#0FFh      ; Isi Register R1 dengan FF  
Delay2: DJNZ   R1,Delay2     ; Kurangi R1 dengan 1, bila hasil belum  
                               ; sama dengan 0 maka lompat ke Delay2  
      DJNZ   R0,Delay1     ; Kurangi R0 dengan 1, bila hasil belum  
                               ; sama dengan 0 maka lompat ke Delay1  
      RET                               ; Kembali ke alamat setelah perintah  
                               ; 'Acall Delay'  
End
```