

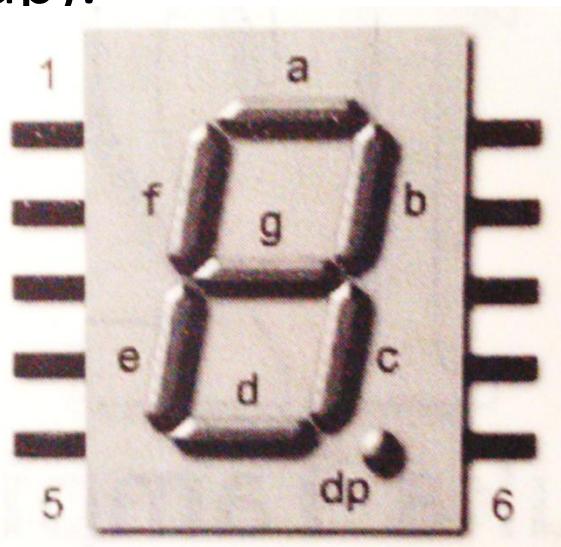
# SISTEM MIKROPROSESOR

ANTARMUKA TAMPILAN LED 7-SEGMENT

Mochamad Fajar Wicaksono, S.Kom., M.Kom.

# PENDAHULUAN

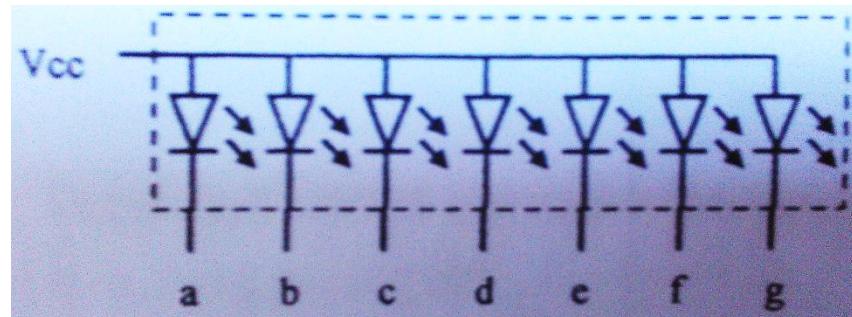
- Tampilan 7 segment pada dasarnya dibuat dari tujuh buah LED yang saling terpisah dan dikemas dalam satu paket.
- Umumnya tampilan 7 segment mempunyai delapan buah LED yang dilengkapi untuk keperluan decimal point (dp).



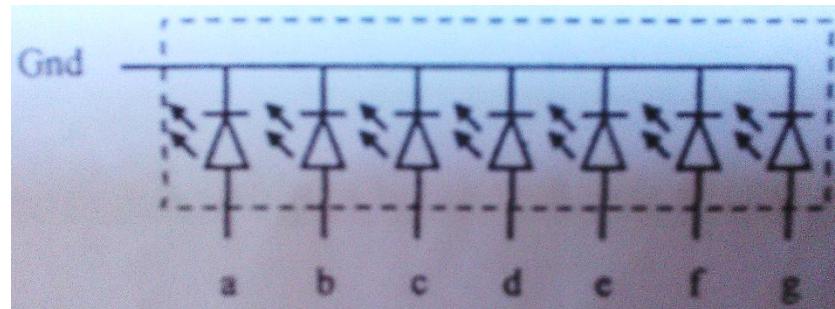
# PENDAHULUAN

Dua tipe tampilan 7 segment:

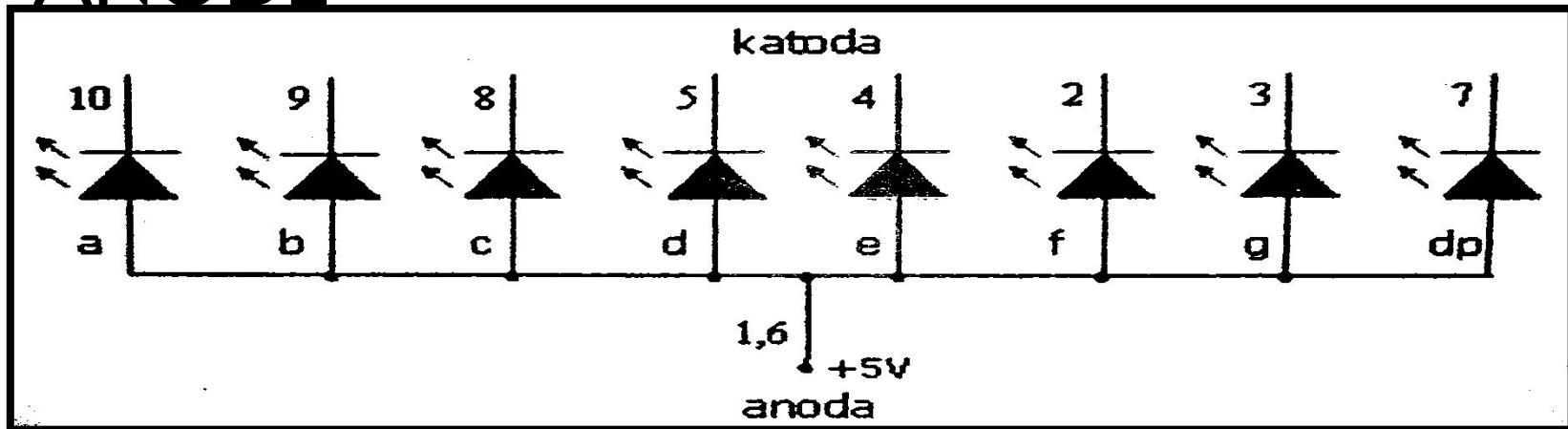
- **Common anode** : semua kaki anoda terhubung ke catu daya +VCC



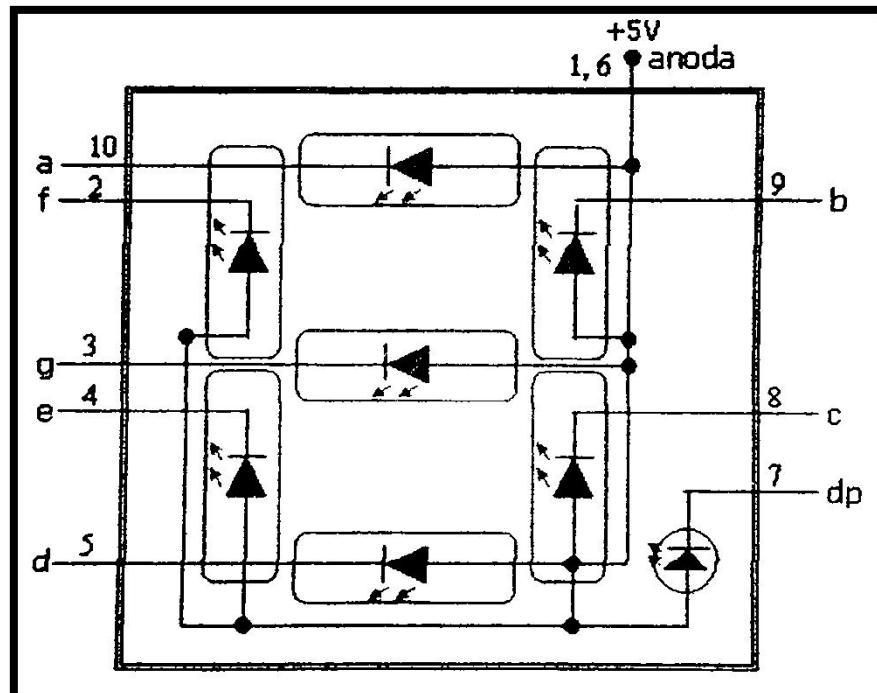
- **Common cathode** : semua kaki katoda terhubung ke ground.



# TAMPAKAN LED 7-SEGMENT COMMON ANODE



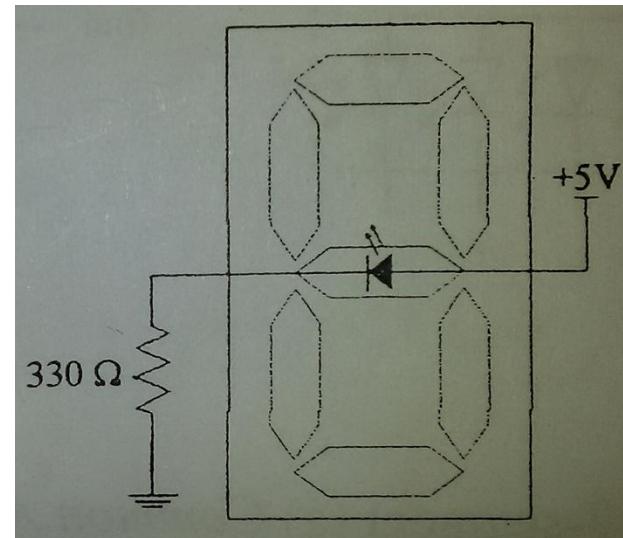
Pada konfigurasi **common anode**, ketujuh pin anode dari LED dihubungkan secara bersama (pada simpul yang sama).



# TAMPAKAN LED 7-SEGMENT COMMON ANODE

- Agar LED dapat memberikan cahaya, maka katoda harus ditanahkan (grounded) melalui hubungan pembatas tahanan seri.
- Untuk mendapatkan pencahayaan LED, nilai tahanan pembatas dapat dicari dengan mengambil jatuh tegangan LED 1,7V dan arus sekitar 10mA.

$$R_{limit} = (5,0 - 1,7)V / 10mA = 330 \text{ Ohm}$$



# TAMPILAN LED 7-SEGMENT COMMON ANODE

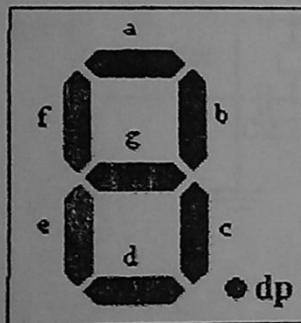
- Tampilan common anode adalah divais yang aktif LOW (LOW-enable) sebab LED mendapatkan tegangan LOW untuk pencahayaan setiap segment.
- IC decoder yang digunakan untuk memberikan drive pada LED harus mempunyai keluaran yang aktif LOW.



# CARA PENGKODEAN LED 7-SEGMENT

- Tampilan LED 7-segment yang digunakan adalah tipe common anode.
- Dengan konfigurasi ini maka LED 7-segment harus diberikan tegangan bersama sebesar +5V sehingga akan segmen akan menyala ketika diberi nilai 0 (LOW) dan akan padam ketika diberi nilai 1 (HIGH).

# CARA PENGKODEAN LED 7-SEGMENT



Tampilan karakter	dp	g	f	e	d	c	b	a	Heksadesimal
0	1	1	0	0	0	0	0	0	C0
1	1	1	1	1	1	0	0	1	F9
2	1	0	1	0	0	1	0	0	A4
3	1	0	1	1	0	0	0	0	B0
4	1	0	0	1	1	0	0	1	99
5	1	0	0	1	0	0	1	0	92
6	1	0	0	0	0	0	1	1	82
7	1	1	1	1	1	0	0	0	F8
8	1	0	0	0	0	0	0	0	80
9	1	0	0	1	0	0	0	0	90
A	1	0	0	0	1	0	0	0	88
b	1	0	0	0	0	0	1	1	83
C	1	1	0	0	0	1	1	0	C6
d	1	0	1	0	0	0	0	1	A1
E	1	0	0	0	0	1	1	0	86
F	1	0	0	0	1	1	1	0	8E

```

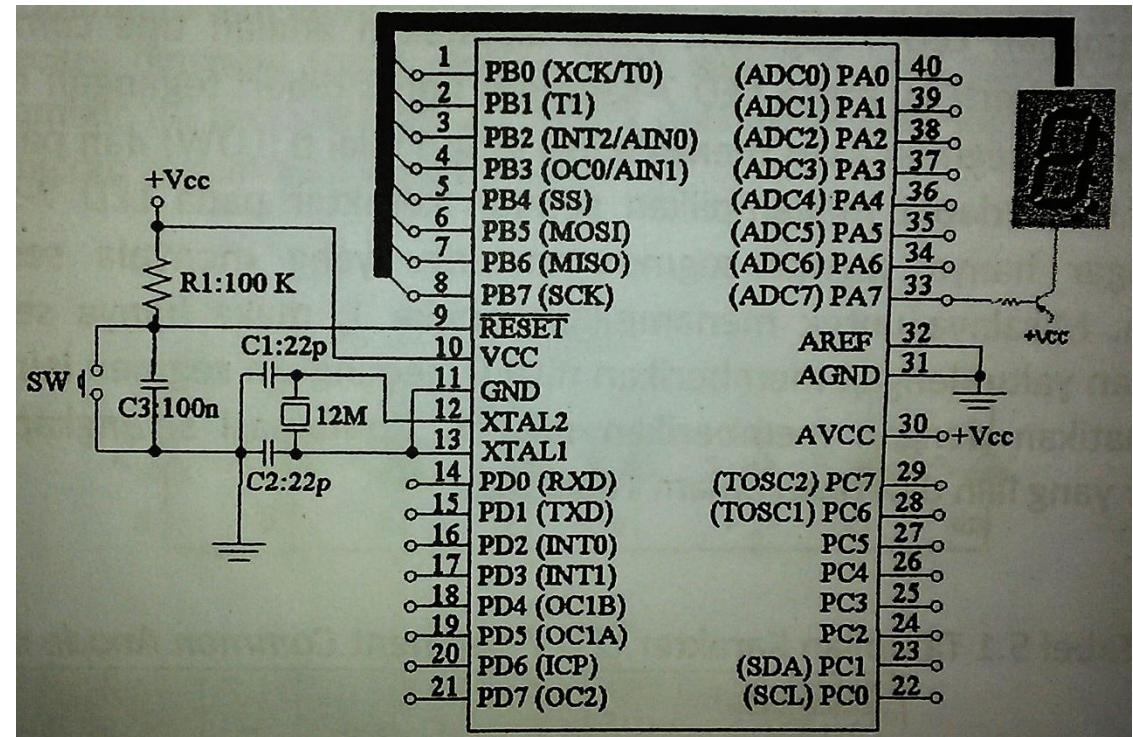
.include "m8535def.inc"
.org 0x0000
rjmp utama
utama:
ldi r16,low(ramend)
out spl,r16
ldi r16,high(ramend)
out sph,r16

ldi r16,0xff
out ddra,r16
out ddrb,r16

ldi r16,0x92
out portb,r16
ldi r16,0b01111111
out porta,r16
stop: rjmp top

```

Program berikut adalah program yang menyalakan angka 5 pada tampilan LED 7-Segment pada port B.



# TEKNIK SCANNING PADA 7-SEGMENT

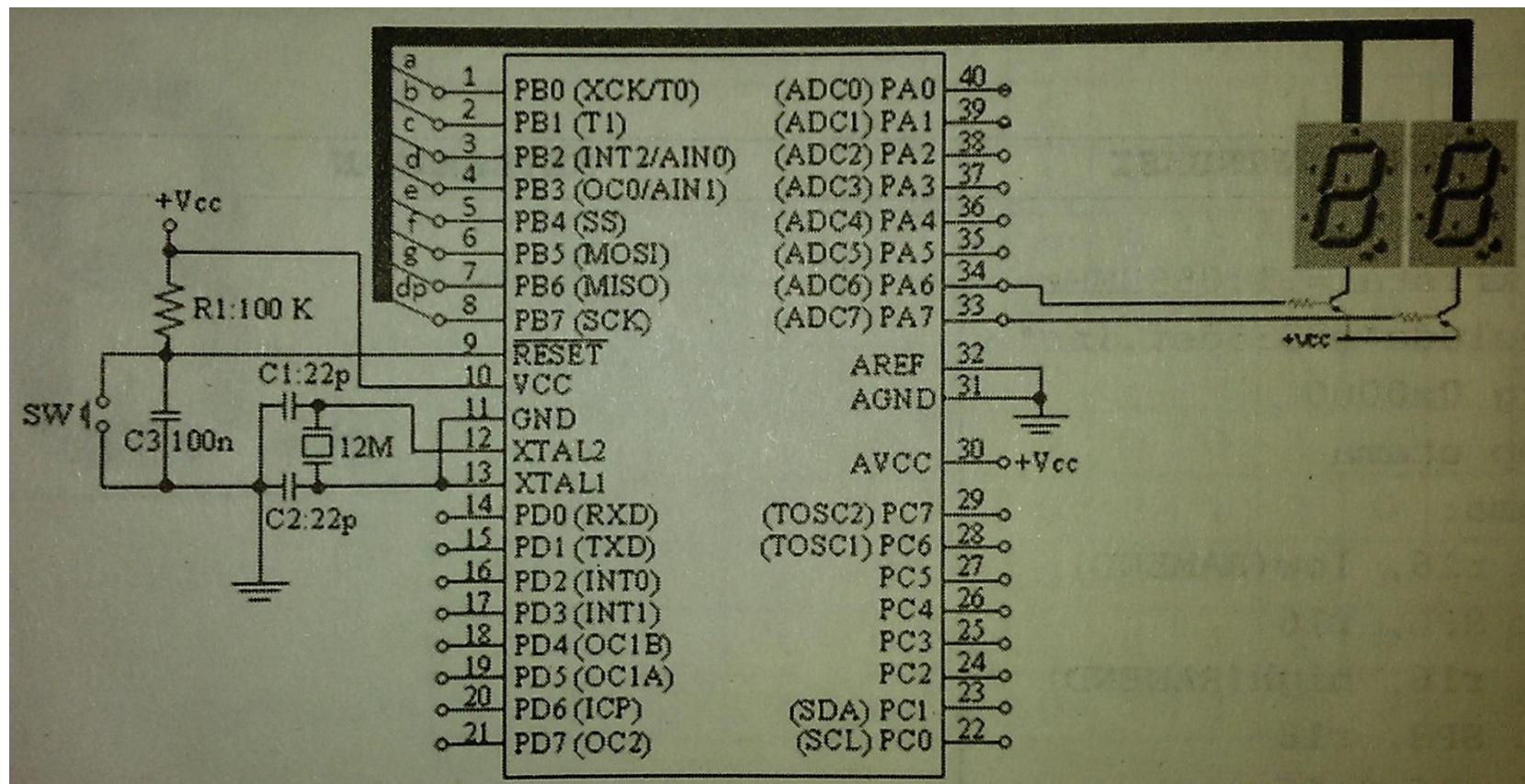
Untuk menampilkan angka yang berderet lebih dari satu digit, misalnya 4 digit maka dapat digunakan port 32 bit tidak efisien.

Untuk mengatasi pemborosan ini dapat digunakan teknik scanning, dimana setiap digit angka atau karakter yang akan ditampilkan dihubungkan secara parallel kemudian ditampilkan secara bergantian.

Teknik scanning ini memanfaatkan kelemahan/ keterbatasan mata manusia yang tidak dapat berakomodasi dengan baik ketika dua atau lebih karakter ditampilkan secara bergantian tetapi dengan delay yang sangat cepat, sehingga seakan-akan karakter tersebut ditampilkan secara simultan atau bersamaan.

Biasanya teknik scanning menggunakan delay dalam orde milidetik misalnya yang biasa digunakan yaitu sekitar 30 sampai 50 milidetik.

# TEKNIK SCANNING PADA 7-SEGMENT



**;=====inisialisasi I/O=====**

```
Ldi r16,0xff  
out ddra,r16  
out ddrb,r16
```

Mulai:

**;=====digit pertama=====**

```
Ldi r16,0xa4  
Out portb,r16  
Ldi r16,0b10111111  
out porta,r16  
Rcall delayscan
```

**;=====digit kedua=====**

```
Ldi r16,0x92  
Out portb,r16  
Ldi r16,0b01111111  
out porta,r16
```

```
Rcall delayscan  
Rjmp mulai
```

Delayscan: Ldi r16,5

Loop1: Ldi r17,25  
Loop2: dec r17

Brne loop2  
Dec r16  
Brne loop1  
ret

Inisialport:

Ldi r16,0xff

Out ddra,r16

Out ddrb,r16

;====counter=====

Lagi2:

Ldi r18,0

Clr r2

;==program utama 7 segment=

Loop:

Ldi zh,high(numerik<<1)

Ldi zl,low(numerik<<1)

Add zl,r18

Adc zh,r2

Lpm

Out portb,r0

Ldi r19,0x00

Out porta,r19

Rcall delay

Inc r18

Cpi r18,10

Brne loop

Rjmp lagi2

Delay: ;delay;

numerik:

.db

0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x8  
2,0xf8,0x80,0x90