

## PRAKTIKUM 7 – SEVEN SEGMENT

**Oleh :** Indra Griha Tofik Isa, M.Kom

### A. Tujuan Pembelajaran

1. Siswa mengetahui fungsi dan kegunaan seven segment
2. Siswa mengetahui seven segment *common anode* dan *common cathode*
3. Siswa dapat membuat rangkaian seven segment penghitung 0 - 9

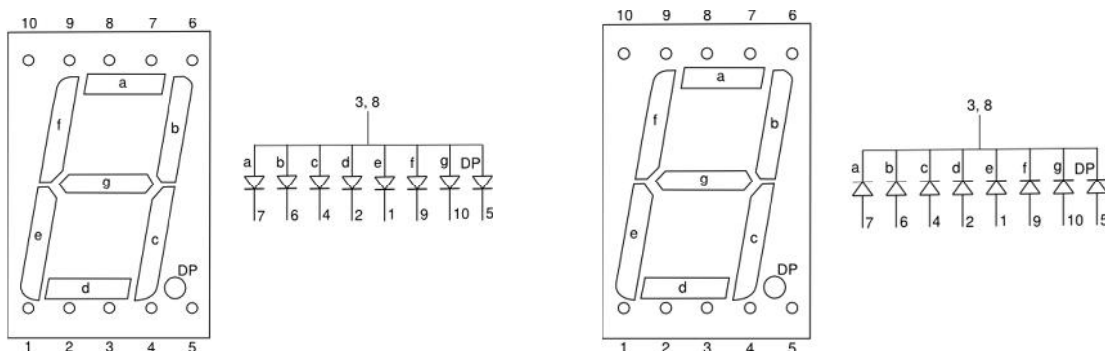
Penampil tujuh segmen biasa digunakan untuk menampilkan digit dengan jumlah digit dengan jumlah 1 sampai dengan 4 digit. Bentuknya dapat dilihat di Gambar 1



**Gambar 1** - Penampil tujuh segmen

Penampil tujuh segmen di bedakan menjadi dua macam: 1) *common anode* dan 2) *common cathode*. Kalau dilihat dari sisi rangkaian, perbedaannya di perlihatkan di Gambar 2. Tampak bahwa:

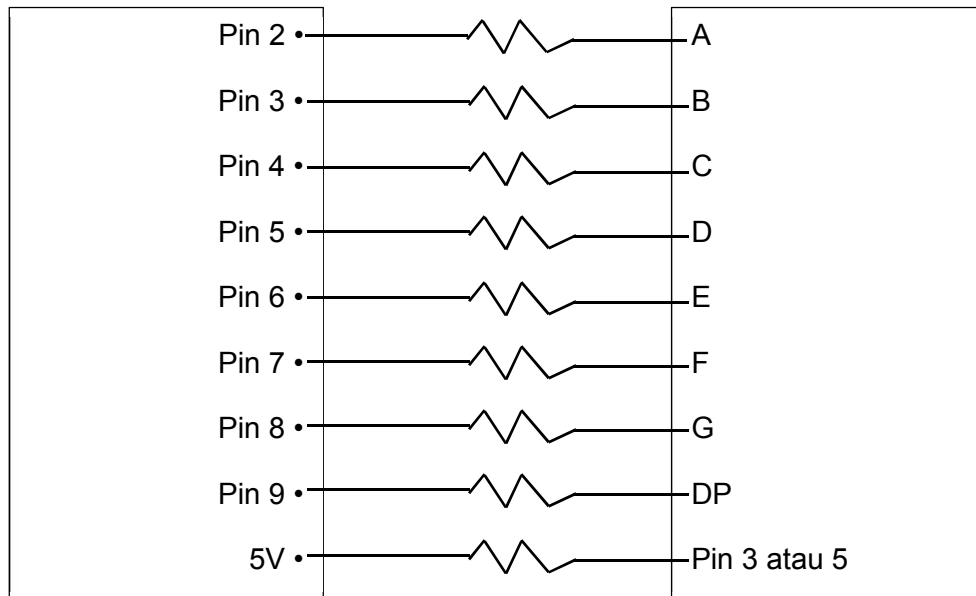
- ✓ Semua anoda dalam keadaan terhubung untuk *common anode*, dan
- ✓ Semua katoda dalam keadaan terhubung untuk *common cathode*.



**Gambar 2** - Perbedaan penampil tujuh segmen untuk *common anode* (kiri) dan *common cathode* (kanan)

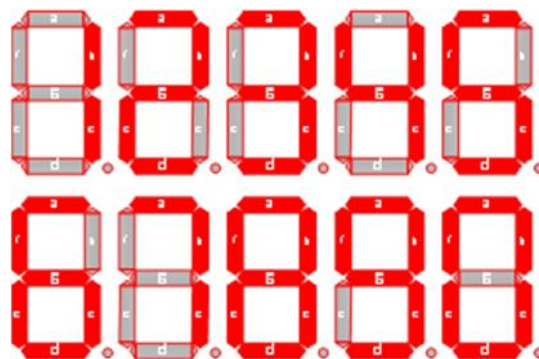
### Menampilkan Digit 0 hingga 9

Untuk mencoba peroyek ini, di perlukan satu penampil tujuh-segmen dan tujuh resistor 220 $\Omega$  atau 1K $\Omega$ . Rangkaiannya terlihat di Gambar 3.



**Gambar 3** - Rangkaian untuk percobaan penampil tujuh-segmen

Sebelum menulis sketch untuk menampilkan bilangan 0 hingga 9, perlu diperhatikan terlebih dahulu susunan setiap angka di penampil tujuh-segmen. Gambar 4 mencantumkan sepuluh angka dan susunan bit untuk setiap segmen. Angka 1 menyatakan setiap segmen menyala dan 0 tidak menyala.



Angka	ABCDEFG dan DP
0	11111100
2	01100000
3	11011010
4	11110010
5	01100110
6	10110110
7	10111110
8	11100000
9	11111110

**Gambar 4** - Digit-digit penampil tujuh-segmen dan bit yang menyusun

Pengujian rangkaian yang melibatkan penampil tujuh-segmen dilakukan dengan menggunakan *sketch* berikut:

### ***Sketch: digit***

```
// -----  
// contoh untuk menampilkan angkatan 0 s/d 9  
//      di penampil tujuh-segmen  
//      khusus untuk common anode  
// -----  
  
const byte ANGKA [10] =  
{  
    B11111100, //0  
    B01100000, //1  
    B11011010, //2  
    B11110010, //3  
    B01100110, //4  
    B10110110, //5  
    B10111110, //6  
    B11100000, //7  
    B11111110, //8  
    B11110110, //9  
};  
const int PIN[8] = {9, 8, 7, 6, 5, 4, 3, 2};  
  
void setup()  
{  
    // Inisialisasi pin  
    for (int j = 0; j < 8; j++)  
        pinMode (PIN[j], OUTPUT);  
}  
void loop()  
{  
    //tampilkan 0 S/D 9  
    for (int j=0; j<10; j++)  
    {  
        tampilkanAngka(j);  
        delay(2000);  
    }  
    delay(2000);  
}  
  
void tampilkanAngka(int angka)  
{  
    // abaikan jika angka tidak antara 0 dan 9  
    if((angka < 0) || (angka > 9))  
        return;  
  
    byte dataSegmen = ANGKA[angka];  
    for (int j=0; j < 8; j++)  
    {  
        int bit = bitRead(dataSegmen, j);  
        if (bit== HIGH)  
            digitalWrite(PIN[j], LOW);  
        else  
            digitalWrite(PIN[j], HIGH);  
    }  
}
```

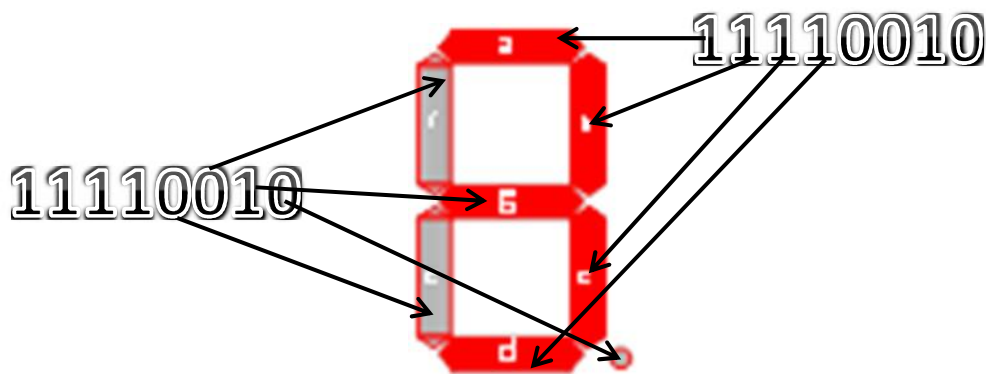
Pada sketch digit penyusun angka di penampil tujuh-segmen didefinisikan di konstanta ANGKA, yang berupa array seperti berikut:

```

const byte ANGKA[10]=
{
B11111100,  //0
B01100000,  //1
B11011010,  //2
B11110010,  //3
B01100110,  //4
B10110110,  //5
B10111110,  //6
B11100000,  //7
B11111110,  //8
B11110110,  //9
};

```

Angka seperti B11110010 menyatakan konstanta biner yang secara berturut-turut menyatakan nilai pada segmen A,B,C,D,E,F,G dan DP sebagai contoh angka tersebut akan menyusun tampilan seperti terlihat di gambar 5



**Gambar 5** bilangan penyusun angka di penampil tujuh segmen

Pertanyaan berikut mendefinisikan konstanta bernama PIN yang menyatakan pin-pin yang digunakan di sketch:

```

const int PIN [8] = {9, 8, 7, 6, 5, 4, 3, 2};

```

Dengan membuat array seperti itu, inisialisasi mode pin dapat ditulis sederhana sebagai berikut:

```

for(int j=0; j < 8; j++)
  pinMode(PIN[j], OUTPUT);

```

Penampilan angka 0 hingga 9 ditangani di loop(). Kodenya:

```

// tampilkan 0 s/d 9
for (int j = 0; j < 10; j++)
{
  tampilkanAngka(j);
  delay (2000);
}

```

Fungsi tampilkanAngka()-lah yang menangani penampilan angka di penampil tujuh-segmen.

Di fungsi tampilkanAngka(), pernyataan berikut akan menolak angka yang kurang dari 0 atau lebih dari 9:

```
if(angka < 0) || (angka > 9))  
    return;
```

Pernyataan berikut digunakan untuk memperoleh data penyusun angka yang akan ditampilkan di penampil tujuh-segmen:

```
byte dataSegmen = ANGKA[angka];
```

Selanjutnya, pemrosesan terhadap angka yang ada di variabel `dataSegmen` untuk diterjemahkan ke angka di penampil tujuh-segmen di tangani oleh:

```
for (int j=0; j<8; j++)  
{  
    int bit = bitRead (dataSegmen, j);  
  
    if (bit == HIGH)  
        digitalWrite(PIN[j],LOW);  
    else  
        digitalWrite(PIN[j],HIGH);  
}
```

Pada pernyataan pengulangan diatas,

```
int bit = bitRead (dataSegmen, j);
```

digunakan untuk memperoleh nilai bit pada bit ke-j. Nilai yang didapat tentu saja berupa 0 (LOW) atau 1 (HIGH). Selanjutnya, nilai HIGH akan diterjemahkan menyetikan segmen dan nilai LOW untuk menghidupkan segmen. Hal penting yang perlu diperhatikan, mengingat penampil tujuh-segmen yang digunakan bersifat common anode, segmen menyala justru jika pin diberi nilai LOW dan mati kalo diberikan nilai HIGH.

Sumber :

Kadir, Abdul. Panduan Mempelajari Aneka Proyek Berbasis Mikrokontroler. 2015. Penerbit ANDI: Yogyakarta

[www.arduino.cc](http://www.arduino.cc)