

MIKROKONTROLER

PROGRAM STUDI
TEKNIK TELKOMUNIKASI
Semester 3

Akuwan Saleh, MT

REFERENSI

- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013.
- James Floyd K & Harold T , “Arduino Adventure Escape from Gemini Station”, Apress, 2013.
- Famosa Studio Arduino Starter Kit Manual – V1.0, Famosa Studio, 2013.
- Martin E, Joshua N, & Jordan H, “Arduino in Action“,Manning Publications.Co, USA, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springe, New York, 2011.
- _____, Sistem minimum Arduino Uno/ATmega328, Instruction Manual, 2010.

MATERI

1. PENDAHULUAN
2. ANTARMUKA MIKROKONTROLER DENGAN LIGHT EMITTING DIODE (LED)
3. ANTARMUKA MIKROKONTROLER DENGAN SAKLAR
4. ANTARMUKA MIKROKONTROLER DENGAN LED DOT MARIK
5. **ANTARMUKA MIKROKONTROLER DENGAN KEYPAD**
6. **ANTARMUKA MIKROKONTROLER DENGAN SEVEN SEGMENT (7-S)**
7. ANTARMUKA MIKROKONTROLER DENGAN LCD 2x16
8. PEMROGRAMAN MELODY
9. ANALOG INPUT (ADC)
10. KOMUNIKASI SERIAL
11. ANTARMUKA MIKROKONTROLER DENGAN LM 35
12. ANTARMUKA MIKROKONTROLER DENGAN LDR
13. ANTARMUKA MIKROKONTROLER DENGAN LAMPU AC 220V
14. ANTARMUKA MIKROKONTROLER DENGAN MOTOR DC
15. ANTARMUKA MIKROKONTROLER DENGAN SENSOR ULTRASONIC
16. Demo Tugas Proyek Semester

5. ANTARMUKA MIKROKONTROLER DENGAN KEYPAD

TUJUAN

- Melakukan proses scanning pada keypad sebagai input
- Dapat membuat proses kontrol display LED sebagai output

DASAR TEORI

Keypad :

- Rangkaian tombol yang berfungsi untuk memberi sinyal pada suatu rangkaian dengan menghubungkan jalur-jalur tertentu.
- Beberapa saklar/tombol, yang terhubung sebagai baris dan kolom, misalnya 3×4 dan 4×4.
- sebagai suatu input pada beberapa peralatan yang berbasis mikroprosesor atau mikrokontroler.

Keypad Matrix 4x4

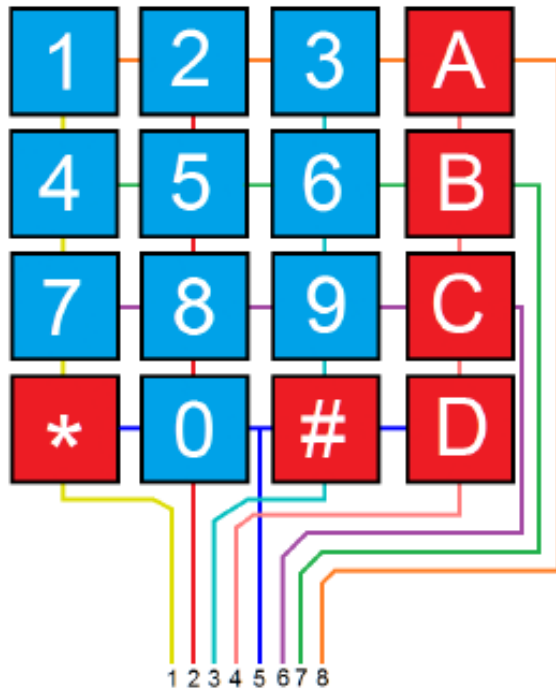
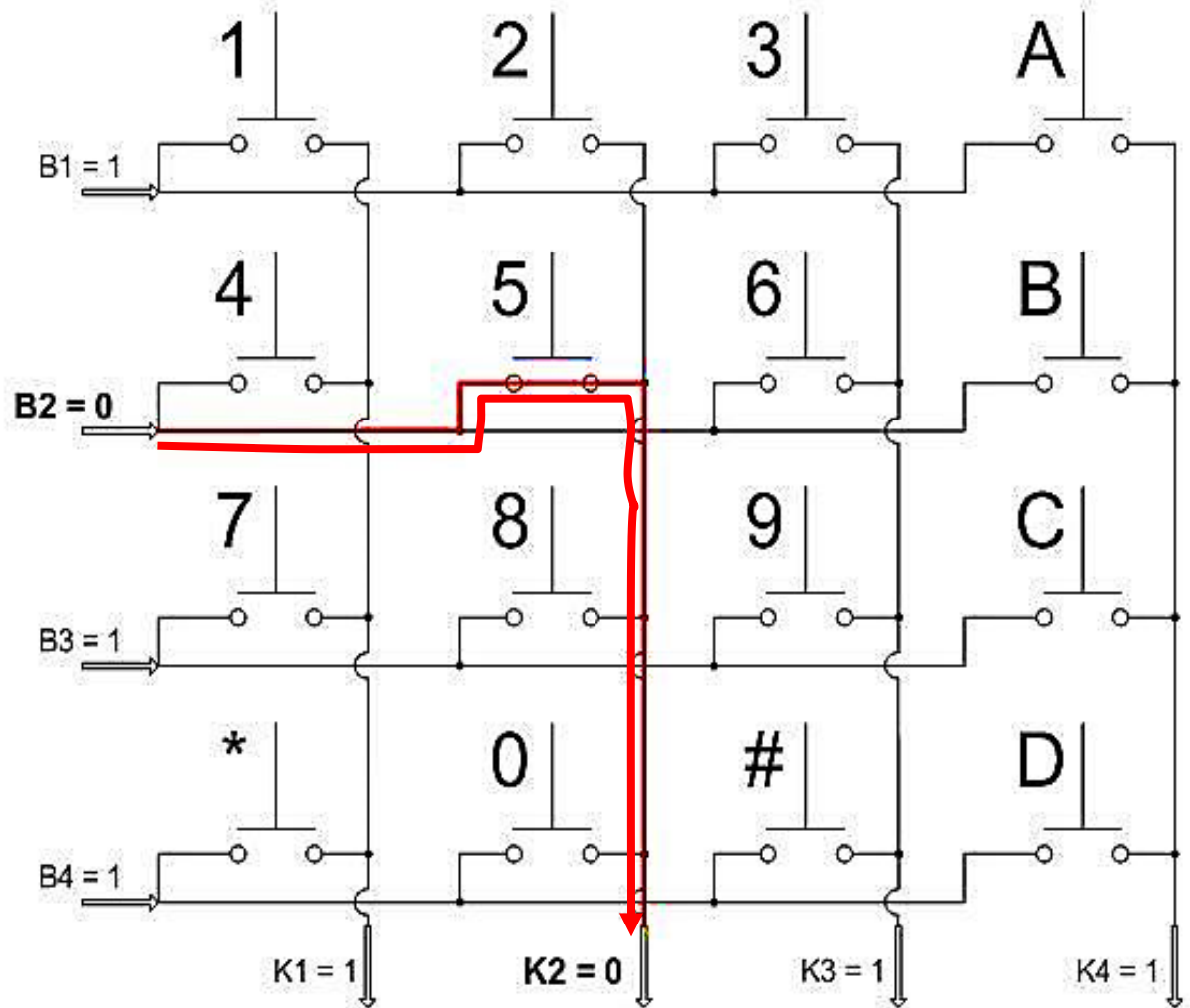


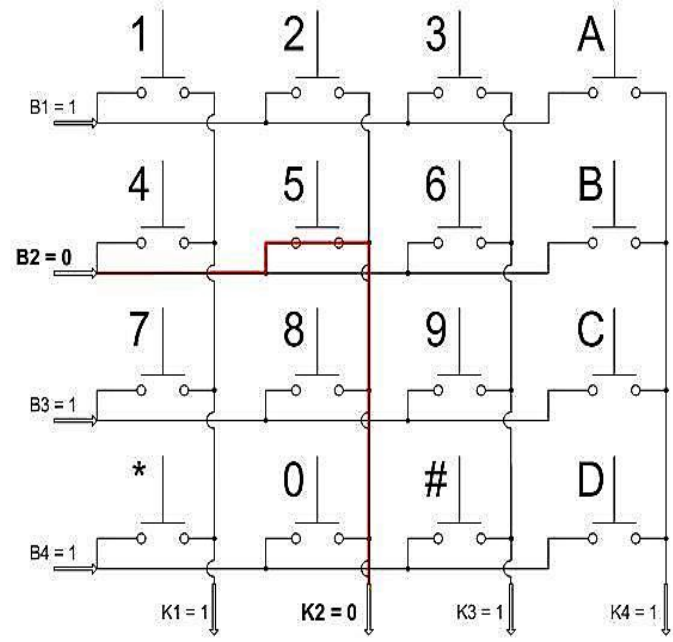
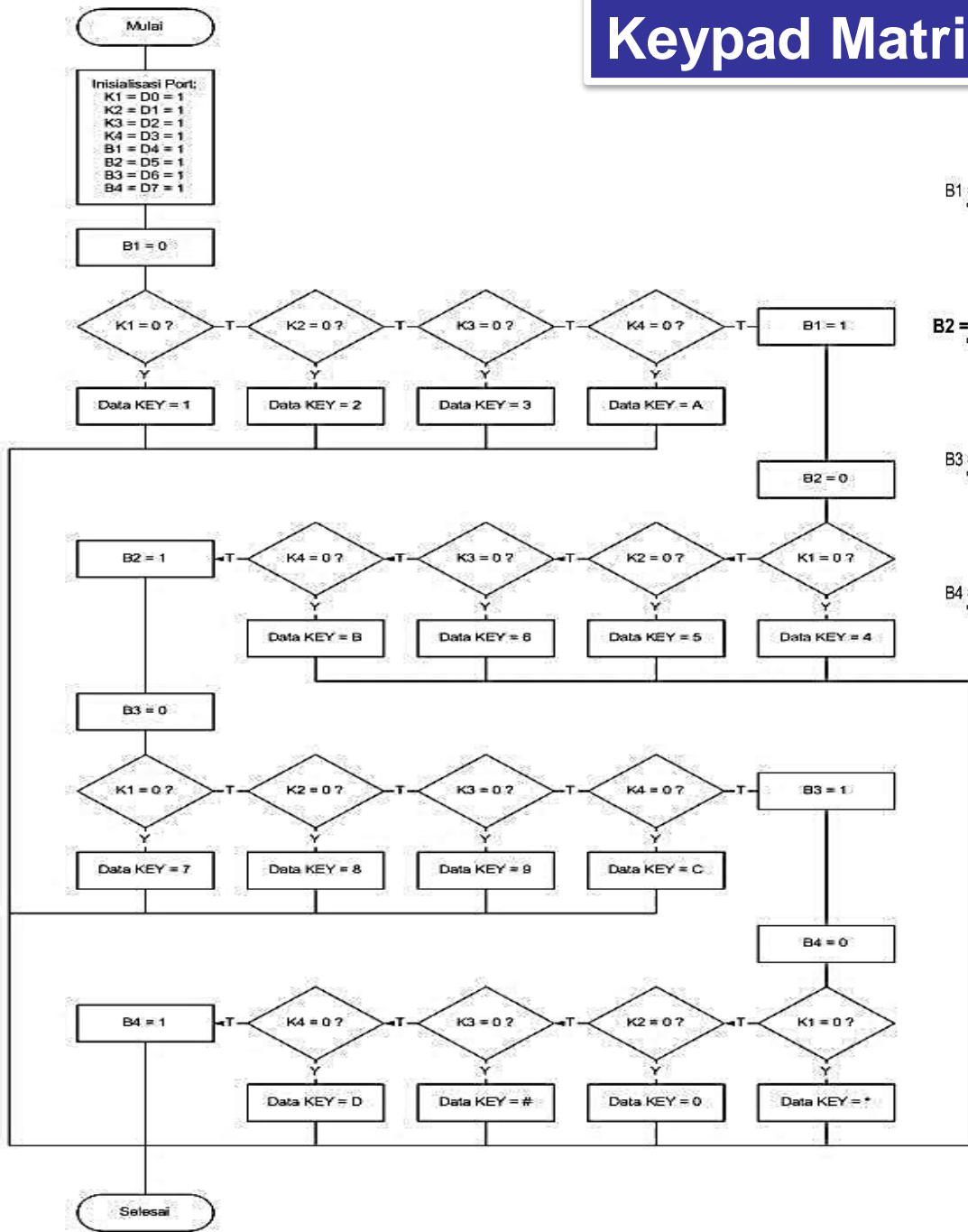
Figure 1: Matrix Keypad Connections

Pin 1-4 = Baris

Pin 5-8 = Kolom



Keypad Matrix 4x4

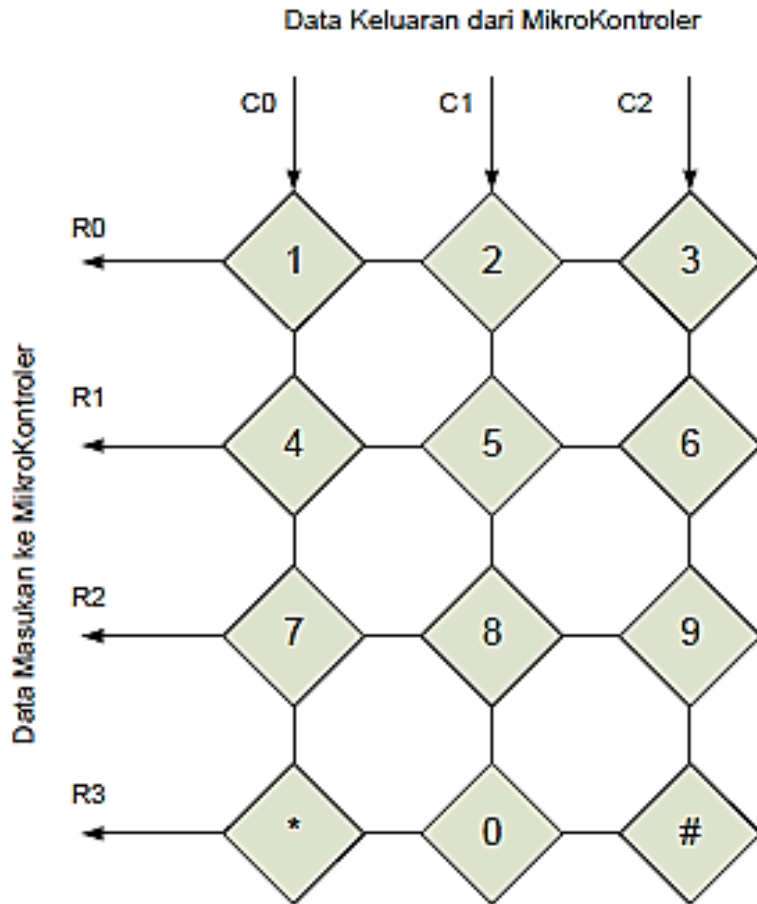


Keypad Matrix 3x4



- Agar mikrokontroler dapat melakukan scan keypad, maka port mengeluarkan salah satu bit dari 3 bit yang terhubung pada kolom dengan logika **low "0"**, selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut.
- Banyak peralatan yang menggunakan keypad, misalnya : *remote control TV*, *Hand phone*, dll.

Scanning Keypad

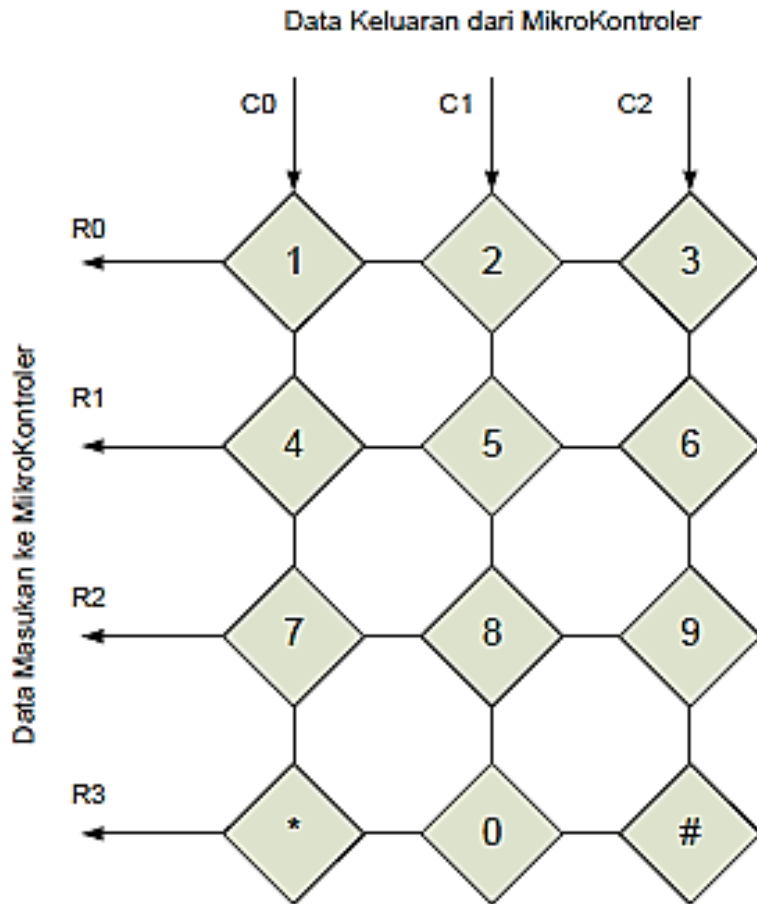


C0-C2 dihubungkan dengan (Output) keluaran dari Mikroprosesor R0-R3 dihubungkan (Input) masukan ke Mikroprosesor

Prinsip Kerja Keypad

- ✓ Proses scanning keypad dilakukan untuk setiap kolom C0-C2 dengan memberikan logika "0" pada kolom yang akan diaktifkan dan memberikan logika "1" pada kolom lainnya.
- ✓ Kemudian membaca data pada baris **R0-R3**. Jika terjadi penekanan tombol pada keypad maka tombol yang ditekan dapat diketahui dengan melihat baris dan kolom yang berlogika "0".
- ✓ Proses meng-aktifkan kolom dilakukan secara bergantian.

Scanning Keypad



- ✓ C0-C2 dihubungkan dengan (Output) keluaran dari Mikroprosesor
- ✓ R0-R3 dihubungkan (Input) masukan ke Mikroprosesor

C0	C1	C2	R0	R1	R2	R3	KEY	Keterangan
0	1	1	0	1	1	1	"1"	Kolom 0 Aktif
0	1	1	1	0	1	1	"4"	
0	1	1	1	1	0	1	"7"	
0	1	1	1	1	1	0	"**"	Kolom 1 Aktif
1	0	1	0	1	1	1	"2"	
1	0	1	1	0	1	1	"5"	
1	0	1	1	1	0	1	"8"	Kolom 2 Aktif
1	0	1	1	1	1	0	"0"	
1	1	0	0	1	1	1	"3"	
1	1	0	1	0	1	1	"6"	Kolom 2 Aktif
1	1	0	1	1	0	1	"9"	
1	1	0	1	1	1	0	"#"	

Hardware :

- Arduino Uno Board
- 1 buah Keypad 3x4
- 1 buah resistor 330 ohm
- 1 buah LED
- Kabel jumper

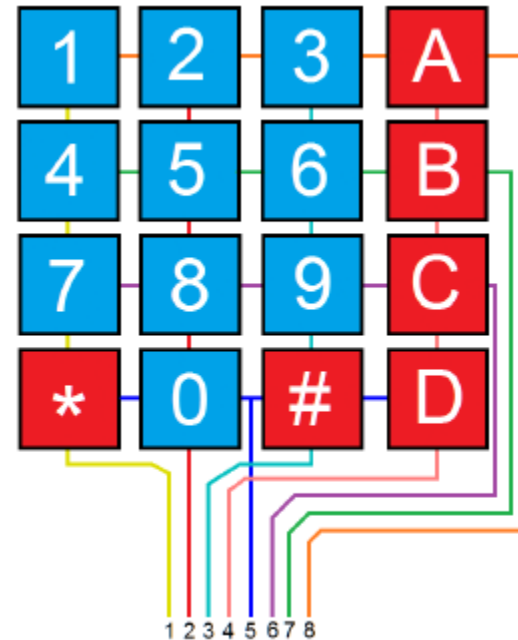


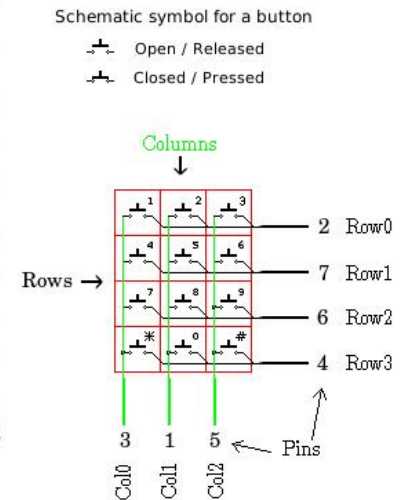
Figure 1: Matrix Keypad Connections

Rangkaian :

Keypad Pin #	Designator	Arduino Pin #
1	Col1	7
2	Row0	5
3	Col0	8
4	Row3	2
5	Col2	6
6	Row2	3
7	Row1	4





- 1: 2+3
- 2: 1+2
- 3: 2+5
- 4: 3+7
- 5: 1+7
- 6: 5+7
- 7: 3+6
- 8: 1+6
- 9: 5+6
- *: 3+4
- 0: 1+4
- #: 5+4

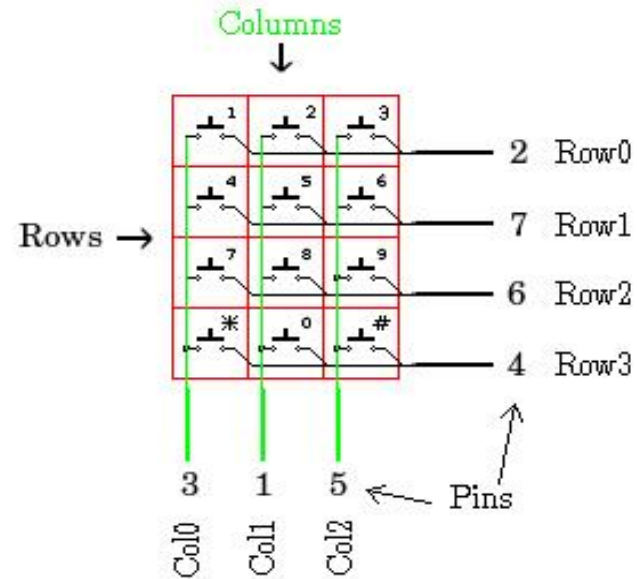




- 1: 2+3
- 2: 1+2
- 3: 2+5
- 4: 3+7
- 5: 1+7
- 6: 5+7
- 7: 3+6
- 8: 1+6
- 9: 5+6
- *: 3+4
- 0: 1+4
- #: 5+4

Schematic symbol for a button

-  Open / Released
-  Closed / Pressed



Penekanan tombol “1” akan menghubungkan pin no. 2 dengan no.3, penekanan tombol “2” akan menghubungkan pin no.1 dengan no.2 dan begitu seterusnya.

PROGRAM-1 :

Keypad 3x4

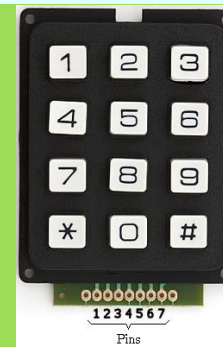
```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
```

```
byte rowPins[ROWS] = {5, 4, 3, 2};
```

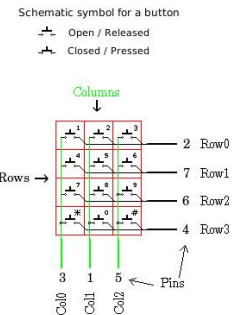
```
byte colPins[COLS] = {8, 7, 6};
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins,
colPins, ROWS, COLS );
```

Keypad Pin #	Designator	Arduino Pin #
1	Col1	7
2	Row0	5
3	Col0	8
4	Row3	2
5	Col2	6
6	Row2	3
7	Row1	4



1: 2+3
2: 1+2
3: 2+5
4: 3+7
5: 1+7
6: 5+7
7: 3+6
8: 1+6
9: 5+6
*: 3+4
0: 1+4
#: 5+4



PROGRAM-1 :

Jika menggunakan Keypad 4x4

Pin Keypad	Pin Arduino
1	5
2	4
3	3
4	2
5	8
6	7
7	6
8	Not Used



Pin 1 Pin 8
pins 1-4 pins 5-8
are the are the
rows columns

Pin-1=baris-1, Pin-2=baris-2
Pin-3=baris-3, Pin-4=baris-4
Pin-5=kolom1, Pin-6=kolom2
Pin-7=kolom3, Pin-8=kolom4

```
byte rowPins[ROWS] = {5, 4, 3, 2};
```

```
byte colPins[COLS] = {8, 7, 6};
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins,  
colPins, ROWS, COLS );
```

PROGRAM 1 - lanjutan

```
byte ledPin = 13;  
boolean blink = false;
```

LED terhubung ke Pin no.13

```
void setup(){  
  Serial.begin(9600);  
  pinMode(ledPin, OUTPUT);  
  digitalWrite(ledPin, HIGH);  
  keypad.addEventListener(keypadEvent); }  
}
```

```
void loop(){  
  char key = keypad.getKey();  
  if (key) {  
    Serial.println(key);  
  }  
  if (blink){  
    digitalWrite(ledPin,!digitalRead(ledPin));  
    delay(100);  
  }  
}
```


PROGRAM 1 - lanjutan

```
void keypadEvent(KeypadEvent key){
  switch (keypad.getState()){
    case PRESSED:
      switch (key){
        case '#':
          digitalWrite(ledPin,!digitalRead(ledPin));
          break;
        case '*':
          digitalWrite(ledPin,!digitalRead(ledPin));
          break;
      }
      break;
    case RELEASED:
      switch (key){
        case '*':
          digitalWrite(ledPin,!digitalRead(ledPin));
          blink = false;
          break;
      }
    }
}
```

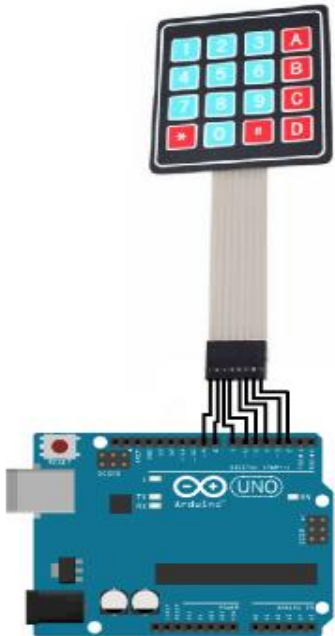
PROGRAM 1 - lanjutan

```
break;  
  case HOLD:  
    switch (key){  
      case '*': blink = true; break;  
    }  
  break;  
}  
}
```

Hardware :

- Arduino Uno Board
- 1 buah Keypad 4x4
- Kabel jumper

Rangkaian :



Keypad Pin	Pin Digital Arduino
Pin 1	Ke pin digital 9
Pin 2	Ke pin digital 8
Pin 3	Ke pin digital 7
Pin 4	Ke pin digital 6
Pin 5	Ke pin digital 5
Pin 6	Ke pin digital 4
Pin 7	Ke pin digital 3
Pin 8	Ke pin digital 2



Pin 1 Pin 8
pins 1-4 pins 5-8
are the are the
rows columns

PROGRAM-2 :

Keypad 4x4

```
#include <Keypad.h>
const byte numRows= 4;
const byte numCols= 4;
char keymap[numRows][numCols]=
{
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
byte rowPins[numRows] = {9,8,7,6};
byte colPins[numCols]= {5,4,3,2};
Keypad myKeypad= Keypad(makeKeymap(keymap),
rowPins, colPins, numRows, numCols);
```

Keypad Pin	Pin Digital Arduino
Pin 1	Ke pin digital 9
Pin 2	Ke pin digital 8
Pin 3	Ke pin digital 7
Pin 4	Ke pin digital 6
Pin 5	Ke pin digital 5
Pin 6	Ke pin digital 4
Pin 7	Ke pin digital 3
Pin 8	Ke pin digital 2

PROGRAM 2 - lanjutan

```
void setup()
```

```
{  
  Serial.begin(9600);  
}
```

Data dilihat di Serial Monitor

```
void loop()
```

```
{  
  char keypressed = myKeypad.getKey();  
  if (keypressed != NO_KEY)  
  {  
    Serial.print(keypressed);  
  }  
}
```

Hasil :

- Catat nyala LED pada pin 13 ketika tombol keypad ditekan dan foto sebagai dokumentasi.
- Buat laporan hasil dari percobaan

Latihan :

1. Modifikasi program-1 supaya dapat menyalakan LED pada pin 13 tidak hanya menggunakan 2 tombol keypad.
2. Buatlah program menyalakan LED menggunakan keypad. Misal: tombol keypad-1 (KP1) untuk menyalakan LED 1, tombol KP2 untuk LED 2 dst.
3. Buat flowchart dari soal latihan no.1 dan 2.

6. ANTARMUKA MIKROKONTROLER DENGAN SEVEN SEGMENT (7-S)

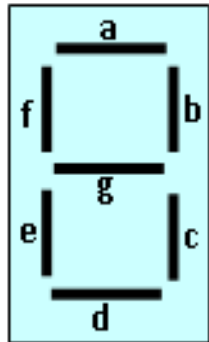
TUJUAN

- Membuat program aplikasi menggunakan media tampilan 7-Segment.
- Membuat proses kontrol display seven segment menggunakan SWITCH sebagai set (H) dan reset (L) program.

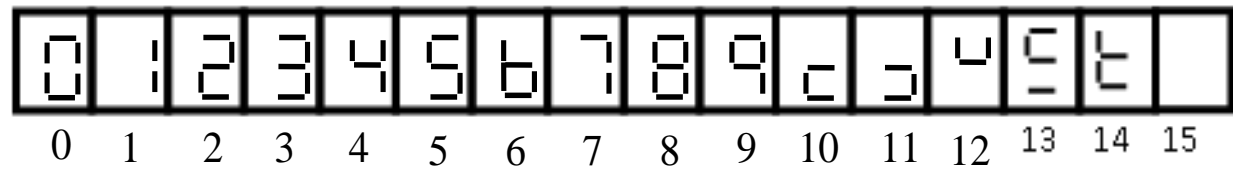
DASAR TEORI

Identifikasi Segment

- Mempermudah pembuatan program untuk menampilkan nyala LED 7-Segment



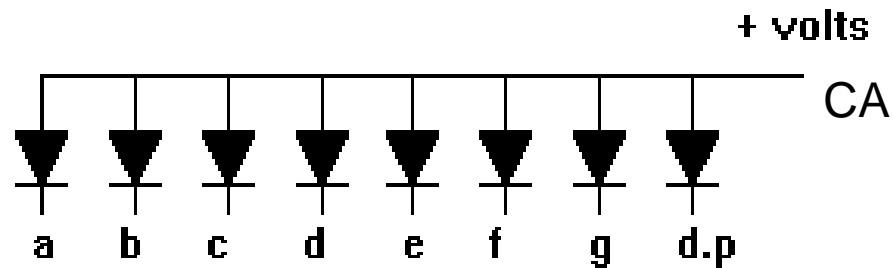
a) Identifikasi segment



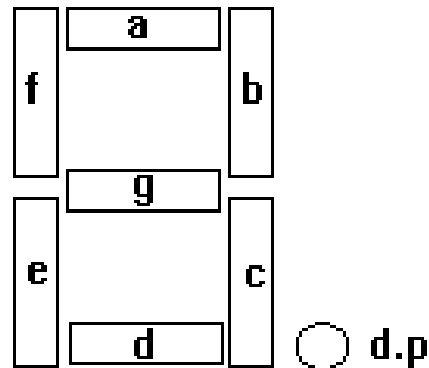
b) Hasil tampilan pada seven segment

- Suatu segment yang digunakan menampilkan angka.

- Kumpulan dari beberapa dioda dengan identifikasi a, b, c, d, e, f, g dan dp (Decimal Point).

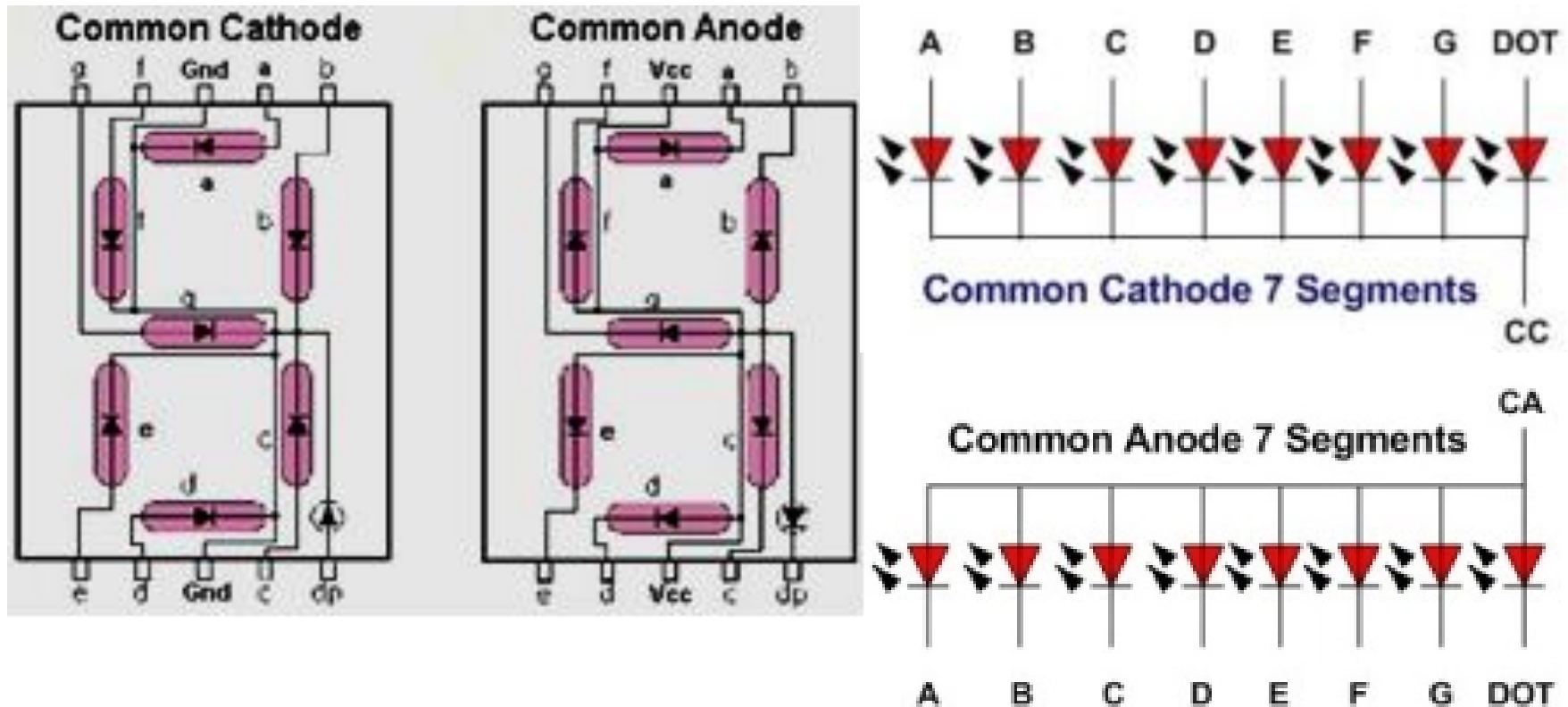


Rangkaian dioda pada 7-Segment



Jenis Seven Segment

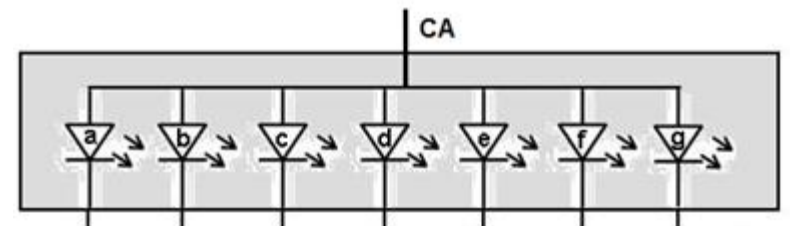
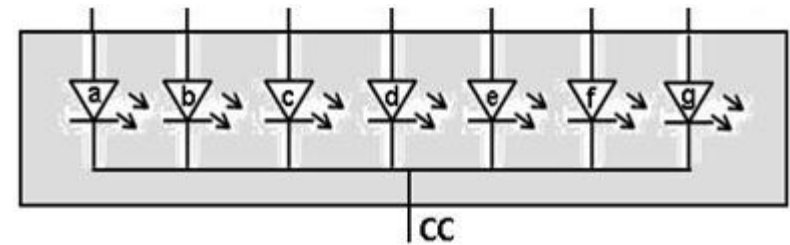
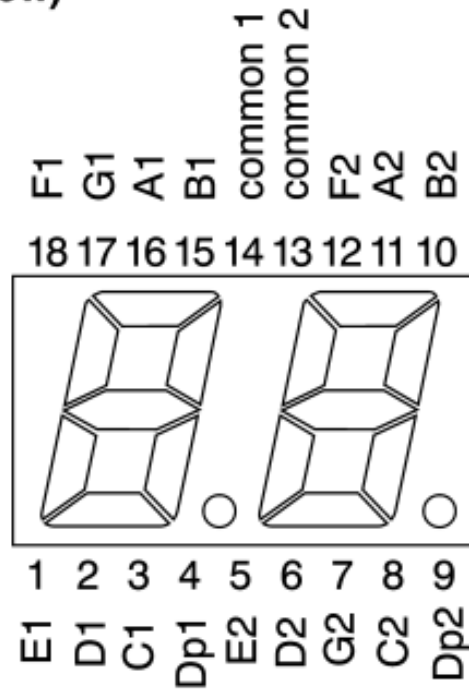
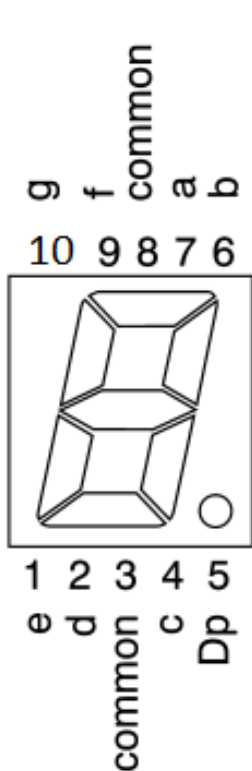
Common Katoda dan Common Anoda



Jenis Seven Segment

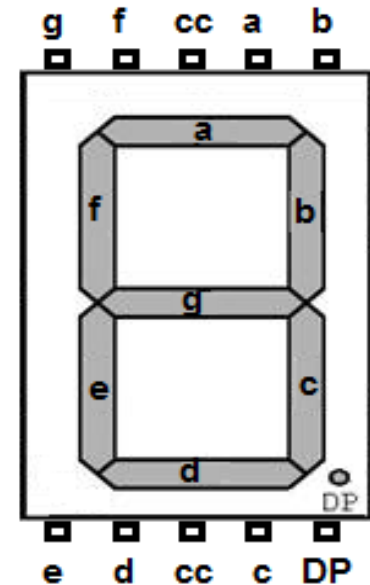
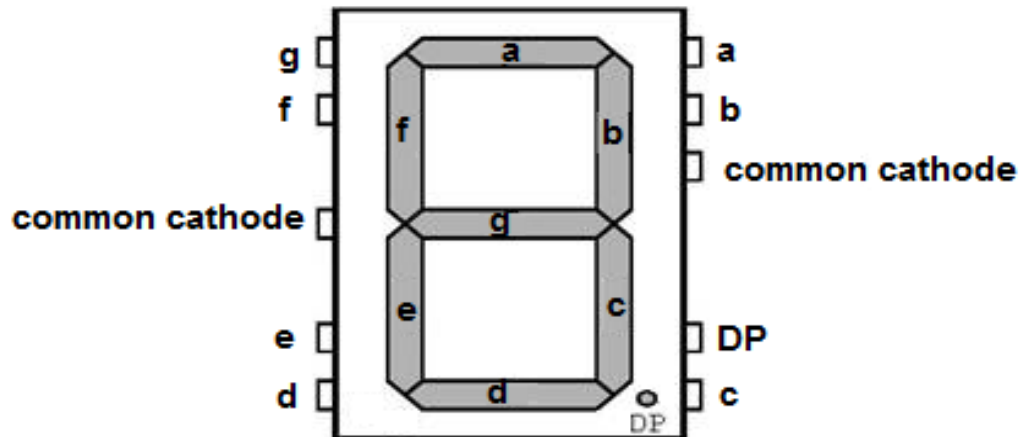
Common Katoda dan Common Anoda

Pin Connectors
(Top View)



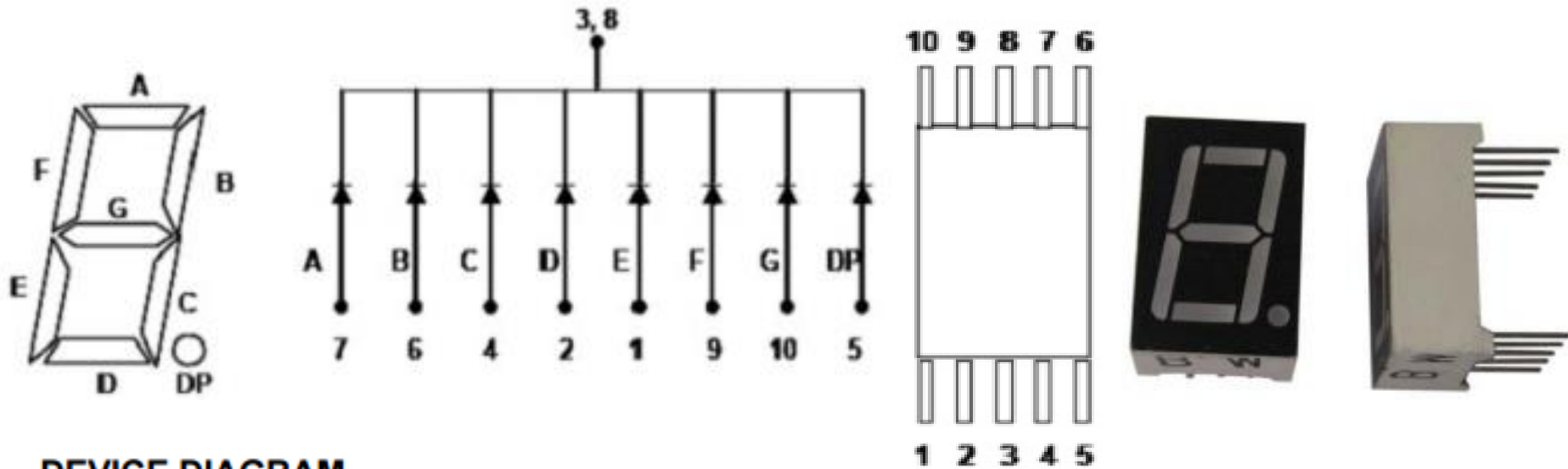
Jenis Seven Segment

Common Katoda



Jenis Seven Segment

Common Cathode 0.56 Inch (14.20mm)

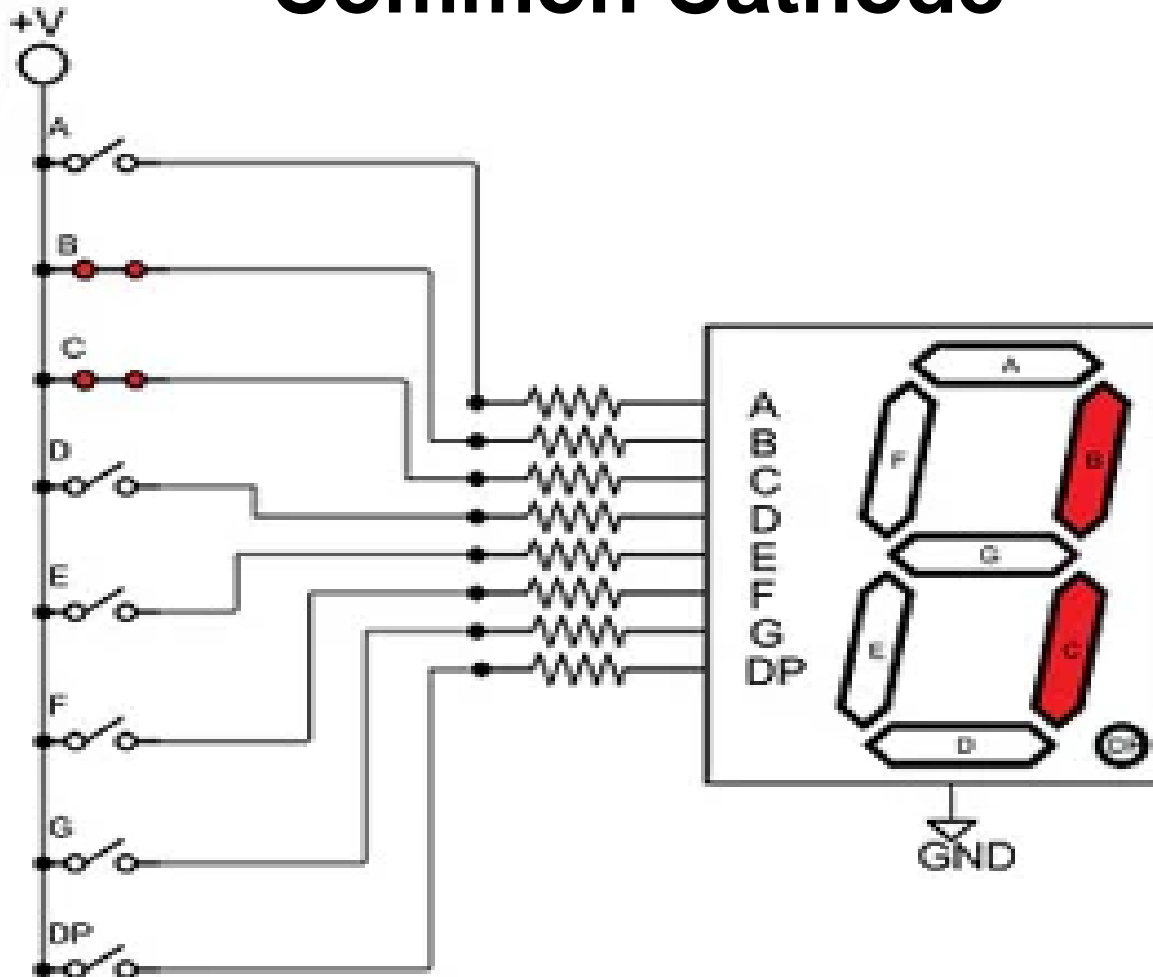


DEVICE DIAGRAM

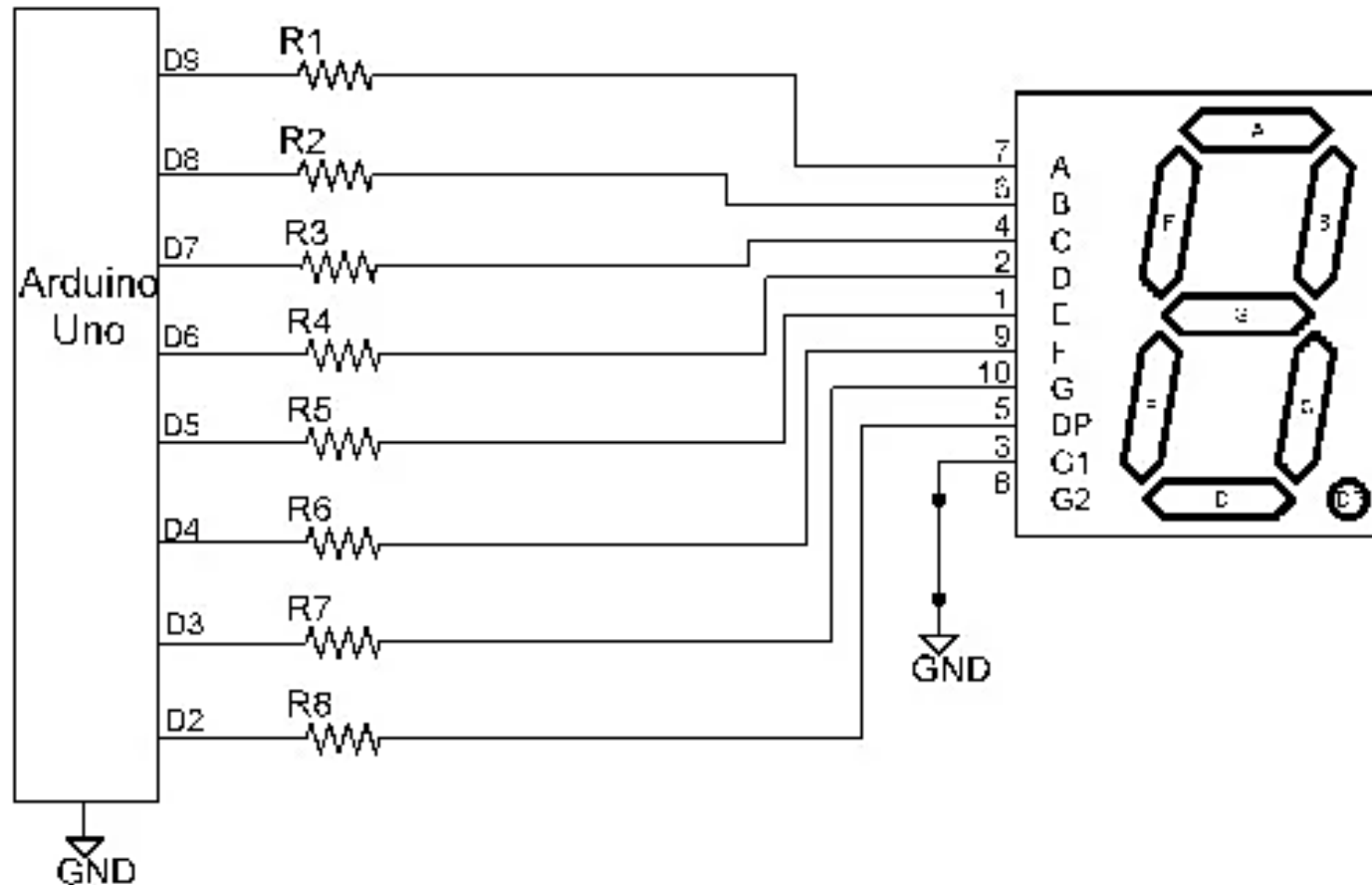
PIN NO.			
1	Anode E	6	Anode B
2	Anode D	7	Anode A
3	Common Cathode DIG. 1	8	Common Cathode DIG. 2
4	Anode C	9	Anode F
5	Anode DP	10	Anode G

Rangkaian Seven Segment

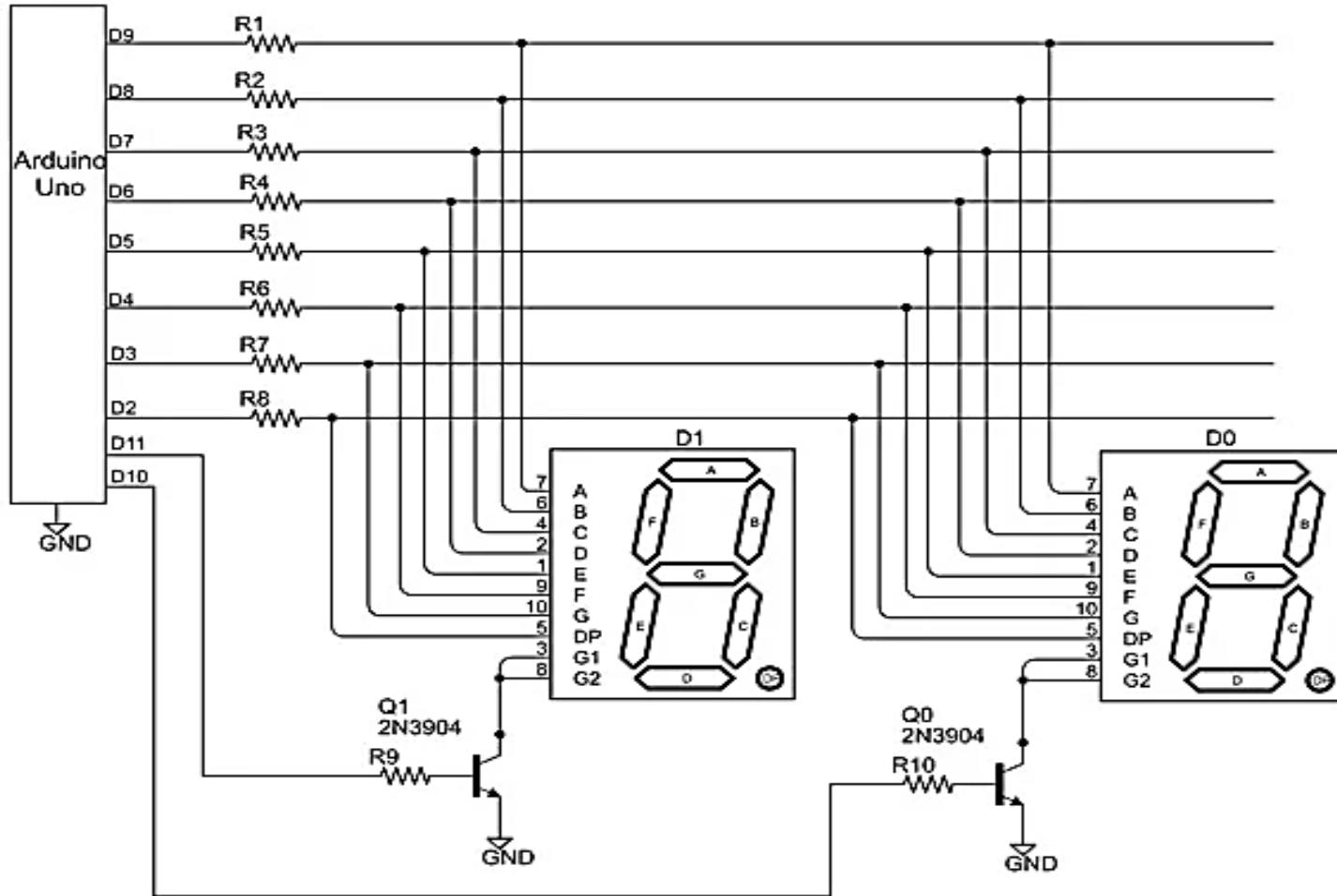
Common Cathode



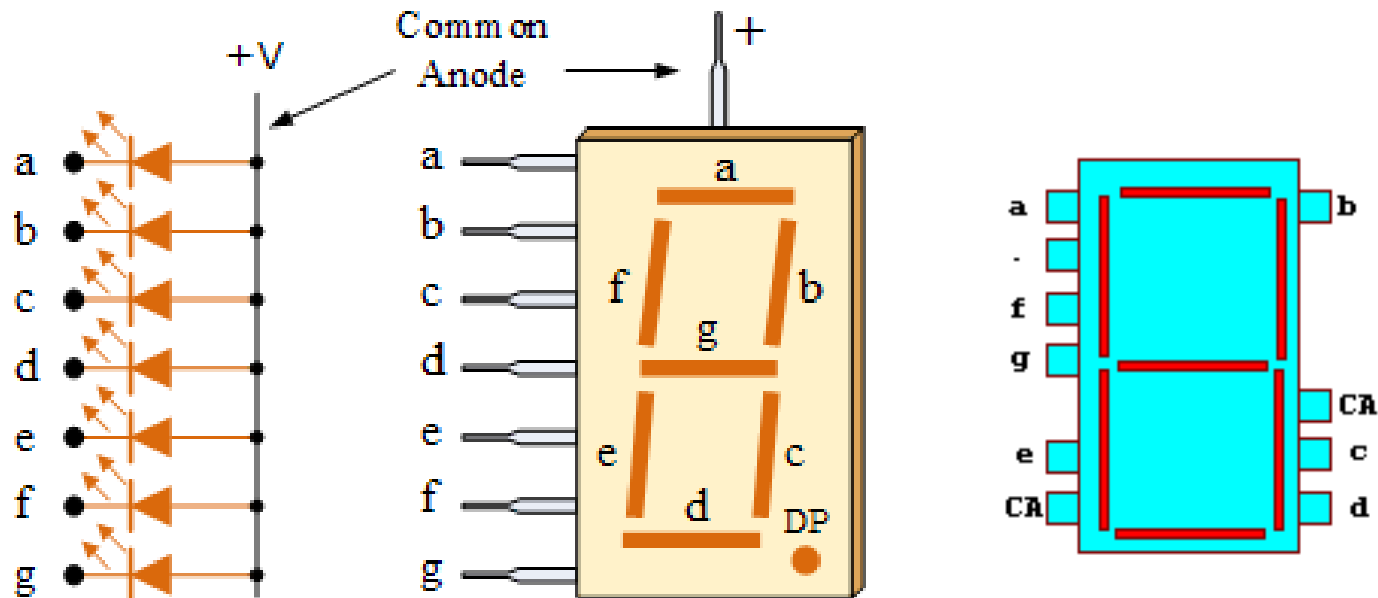
Rangkaian Seven Segment Common Cathode



Rangkaian Seven Segment Common Cathode

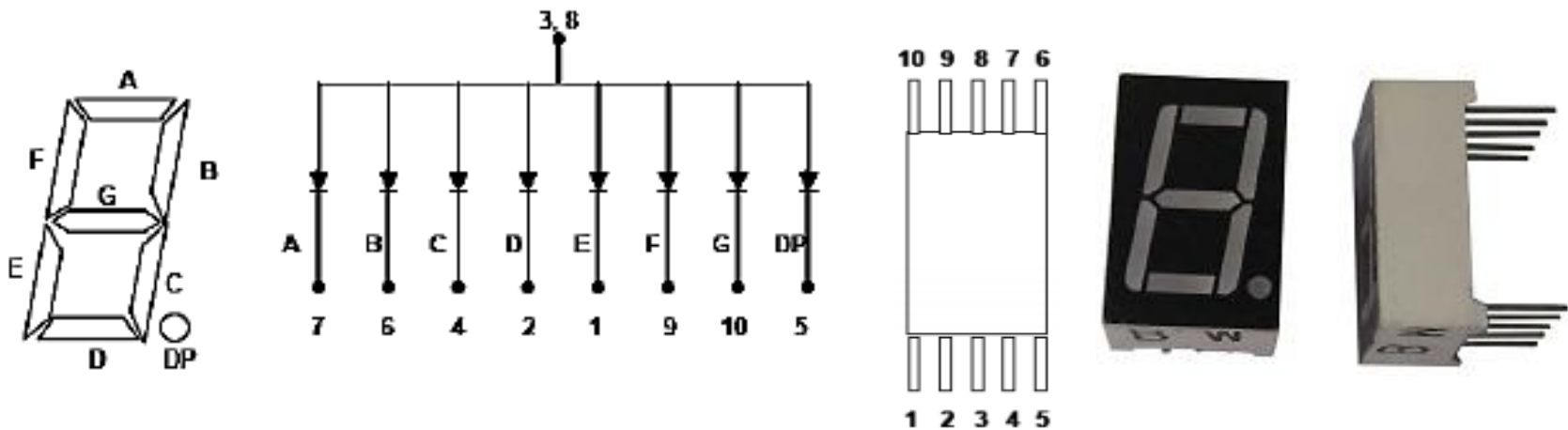


Jenis Seven Segment Common Anoda



Jenis Seven Segment

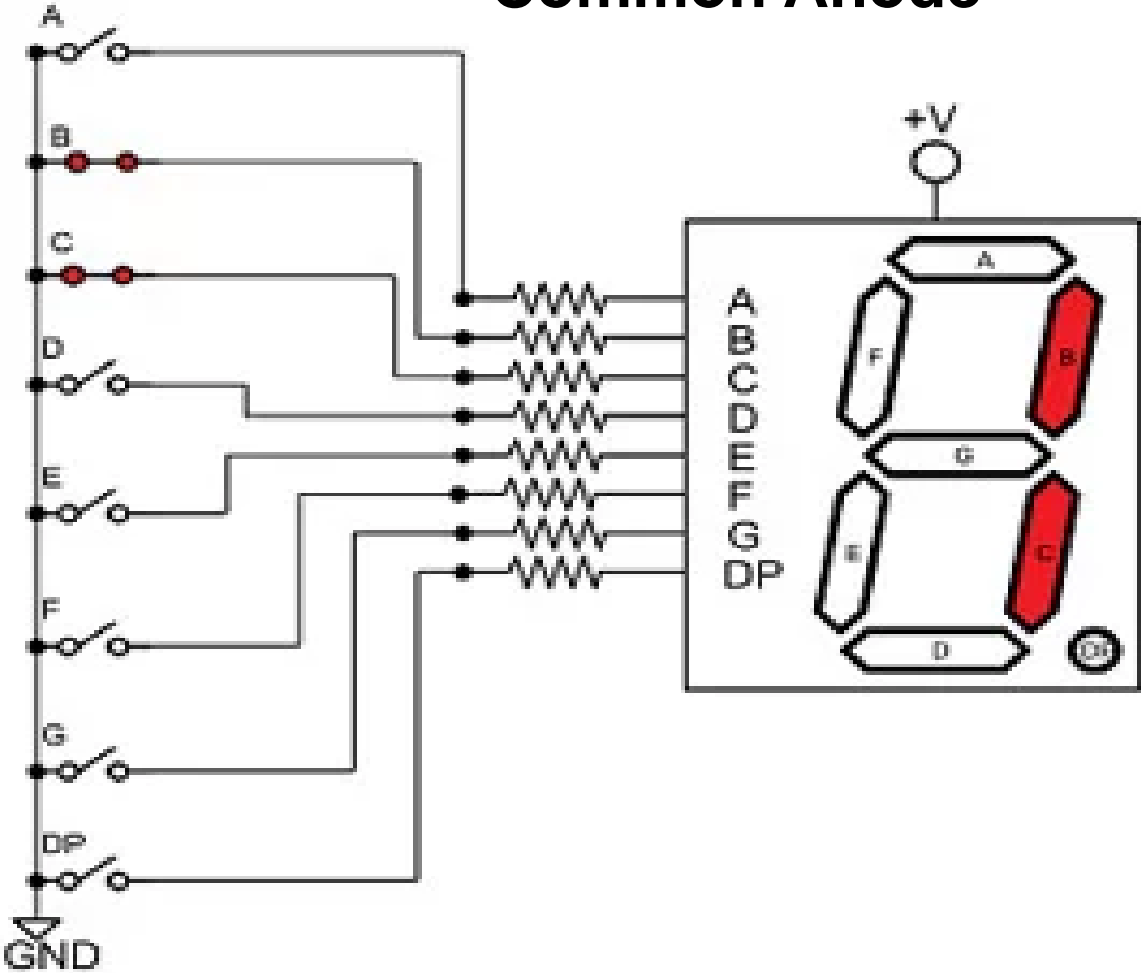
Common Anode 0.56 Inch (14.20mm)



PIN NO.			
1	Cathode E	6	Cathode B
2	Cathode D	7	Cathode A
3	Common Anode DIG. 1	8	Common Anode DIG. 2
4	Cathode C	9	Cathode F
5	Cathode DP	10	Cathode G

Rangkaian Seven Segment

Common Anode

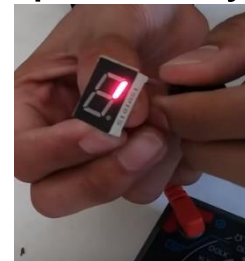


Jenis 7-S

Menggunakan Multitester

Common Anoda

1. kalibrasi multitester pada skala 1 ohm
2. tempatkan penyindik hitam (-) pada multitester ke 7S bagian tengah (CA).
3. tempatkan penyindik merah (+) pada multitester ke 7S bagian yang lain secara berpindah pindah maka akan menyalakan lampu tiap bagian pada 7S begitu seterusnya sampai lampu segment dapat menyala secara berurutan.



Common Katoda

1. kalibrasi multitester pada skala 1 ohm
2. tempatkan penyindik merah (+) pada multitester ke 7S bagian tengah (CC).
3. tempatkan penyindik hitam (-) pada multitester ke 7 segment bagian kawat yang lain secara berpindah pindah maka akan menyalakan lampu tiap bagian pada 7S begitu seterusnya sampai lampu segment dapat menyala secara berurutan

1. Common Anoda

- Semua anoda LED seven segmen disatukan dihubungkan ke VCC, kemudian LED dihubungkan melalui tahanan pembatas arus keluar dari penggerak LED.
- COMMON ANODA berada pada kondisi AKTIF LOW (led akan menyala/aktif bila diberi logika 0).

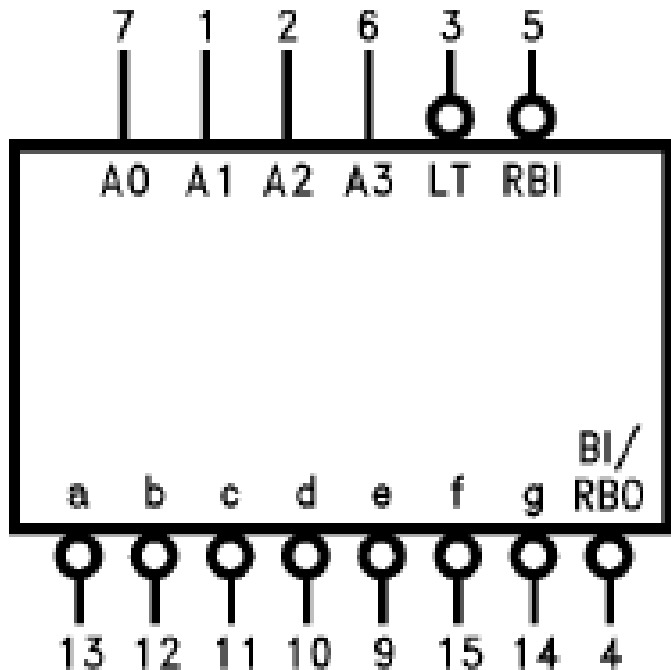
2. Common Katoda

- Semua katoda disatukan dihubungkan ke GROUND.
- COMMON KATODA berada pada kondisi AKTIF HIGH (led akan menyala/aktif bila diberi logika 1).

Prinsip Kerja 7S dgn Driver

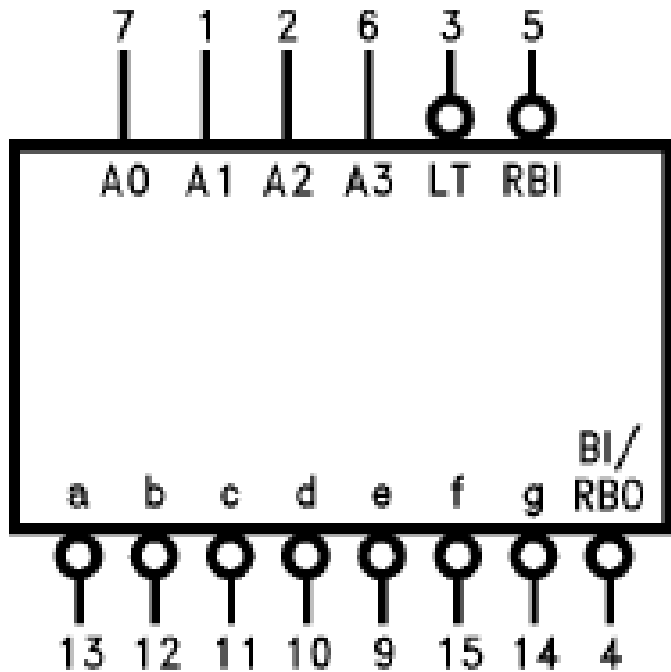
- Input biner dimasukkan ke dalam decoder, kemudian decoder mengkonversi bilangan biner tersebut menjadi decimal, yang akan ditampilkan pada 7 segment.
- Sebuah decoder atau seven segment driver akan mengatur aktif tidaknya led-led dalam seven segment sesuai dengan nilai biner yang diberikan.
- Dekoder BCD ke 7 segment :
 - menyalakan 7 segment mode common anoda IC TTL 74LS47
 - menyalakan 7 segment mode common katoda IC TTL 74LS48, CD4511.

Konfigurasi pin IC 74LS47



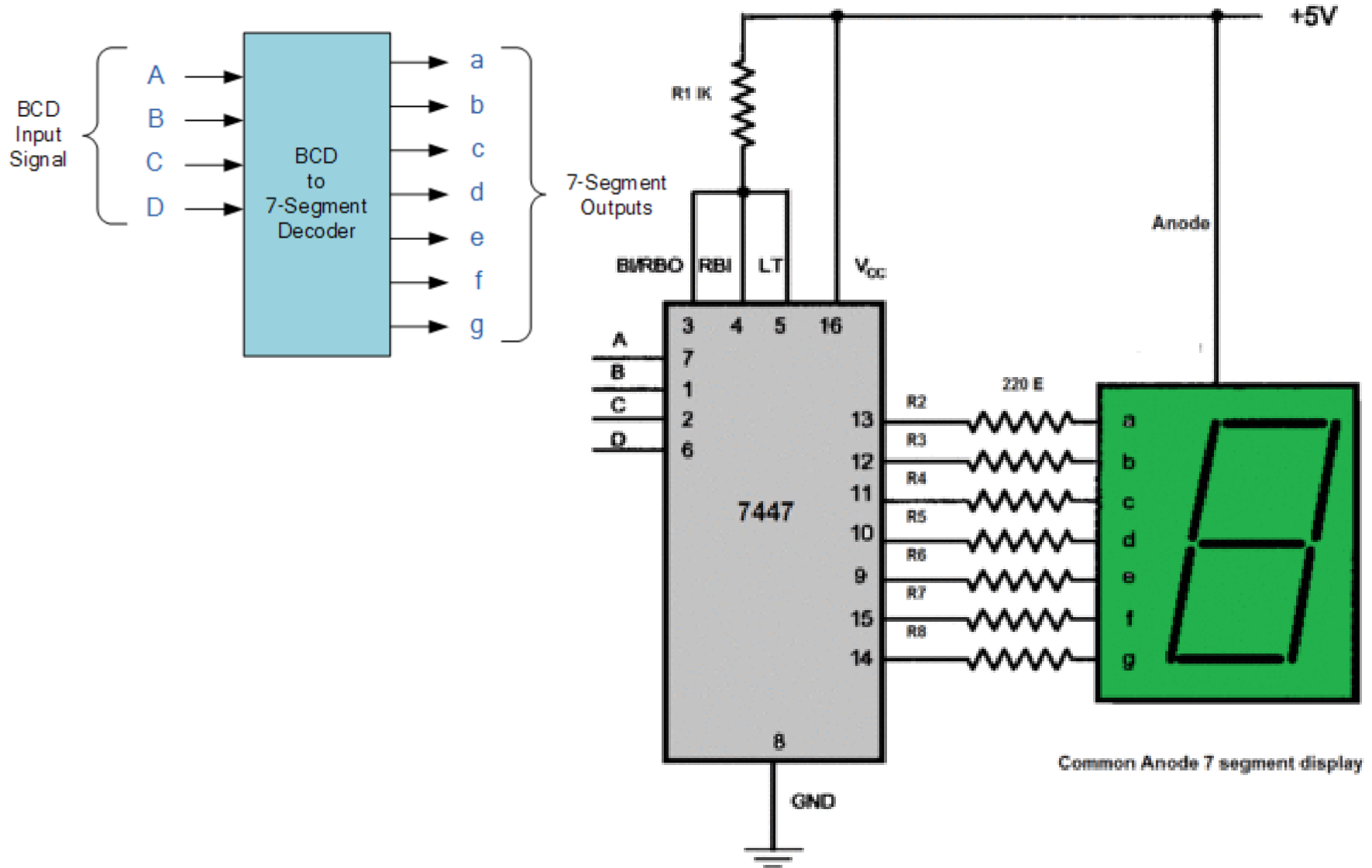
- Dekoder BCD ke 7 segment mempunyai masukan berupa bilangan BCD 4-bit (masukan A0, A1, A2 dan A3).
- Bilangan BCD ini dikodekan sehingga membentuk kode tujuh segmen yang akan menyalakan ruas-ruas yang sesuai pada seven segment.
- Masukan BCD diaktifkan oleh logika '1', dan keluaran dari dekoder 7447 adalah aktif low.
- Tiga masukan ekstra juga ditunjukkan pada konfigurasi pin IC 7447 yaitu lamp test, blanking input/ripple blanking output, dan ripple blanking input.

Konfigurasi pin IC 74LS47

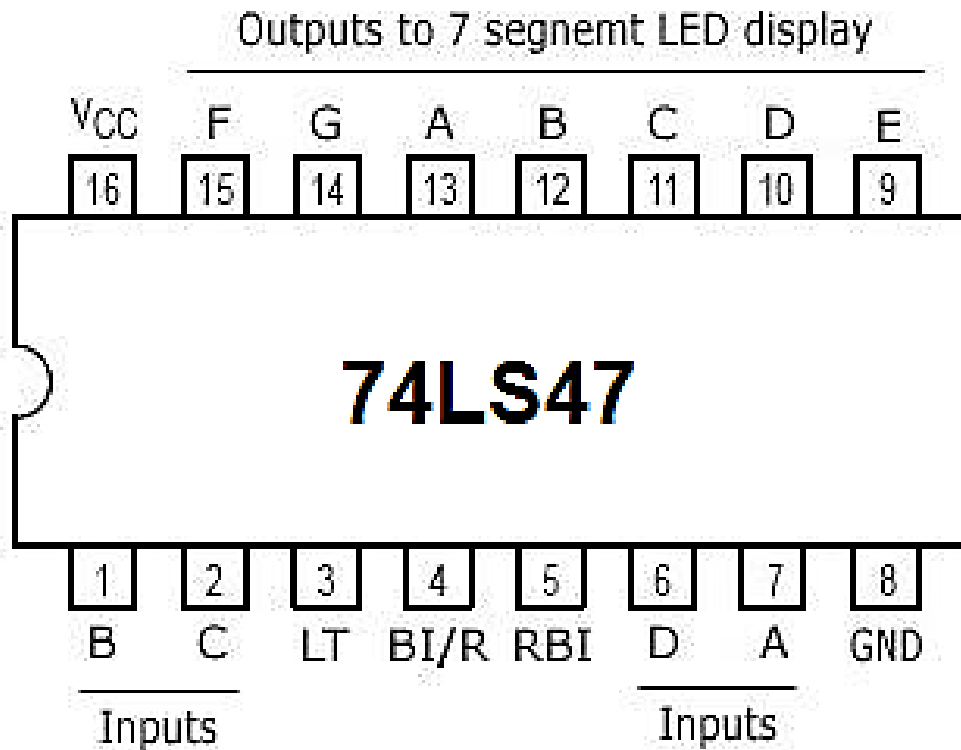


- **Lamp Test**, berfungsi mengeset display, bila diberi logika '0' maka semua keluaran IC = berlogika 0 dan Seven segment = angka delapan (8).
- **BI/RBO'**, berfungsi mematikan keluaran dari IC. Bila diberi logika "0" maka semua keluaran IC = berlogika "1" dan seven segment akan mati.
- **RBI'**, berfungsi mematikan keluaran dari IC jika semua input berlogika "0". Bila diberi logika "0", maka semua keluaran IC akan berlogika "1" dan seven segment akan mati.

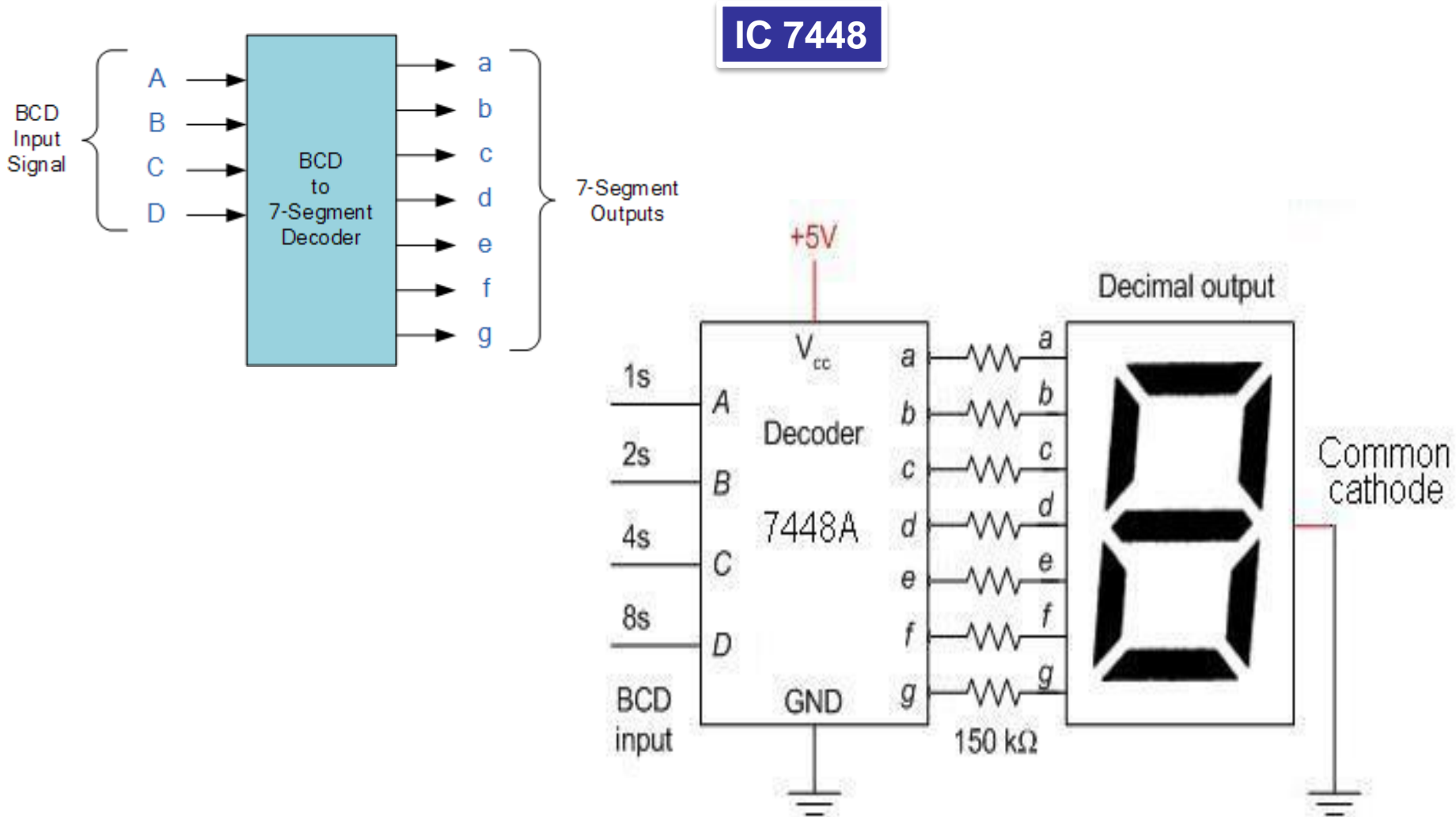
Driver Common Anoda



Driver Common Anoda

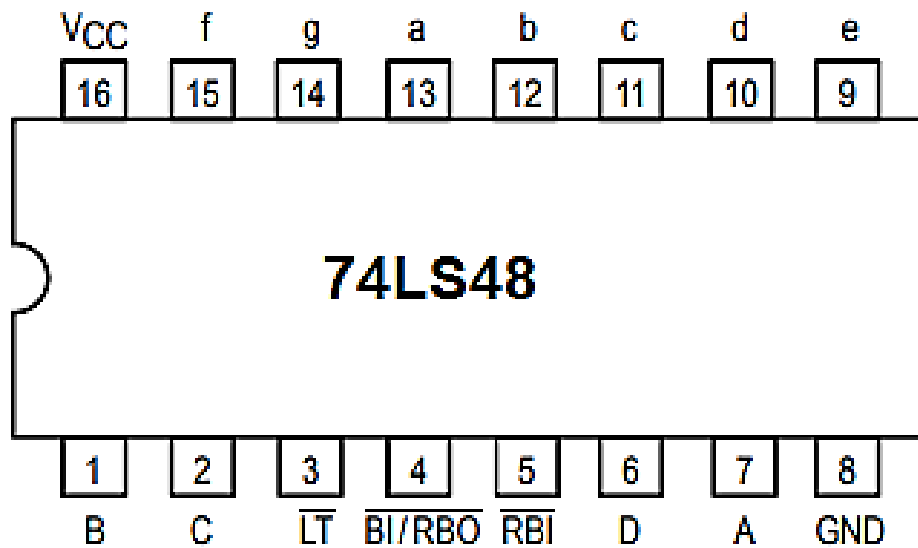


Driver Common Katoda

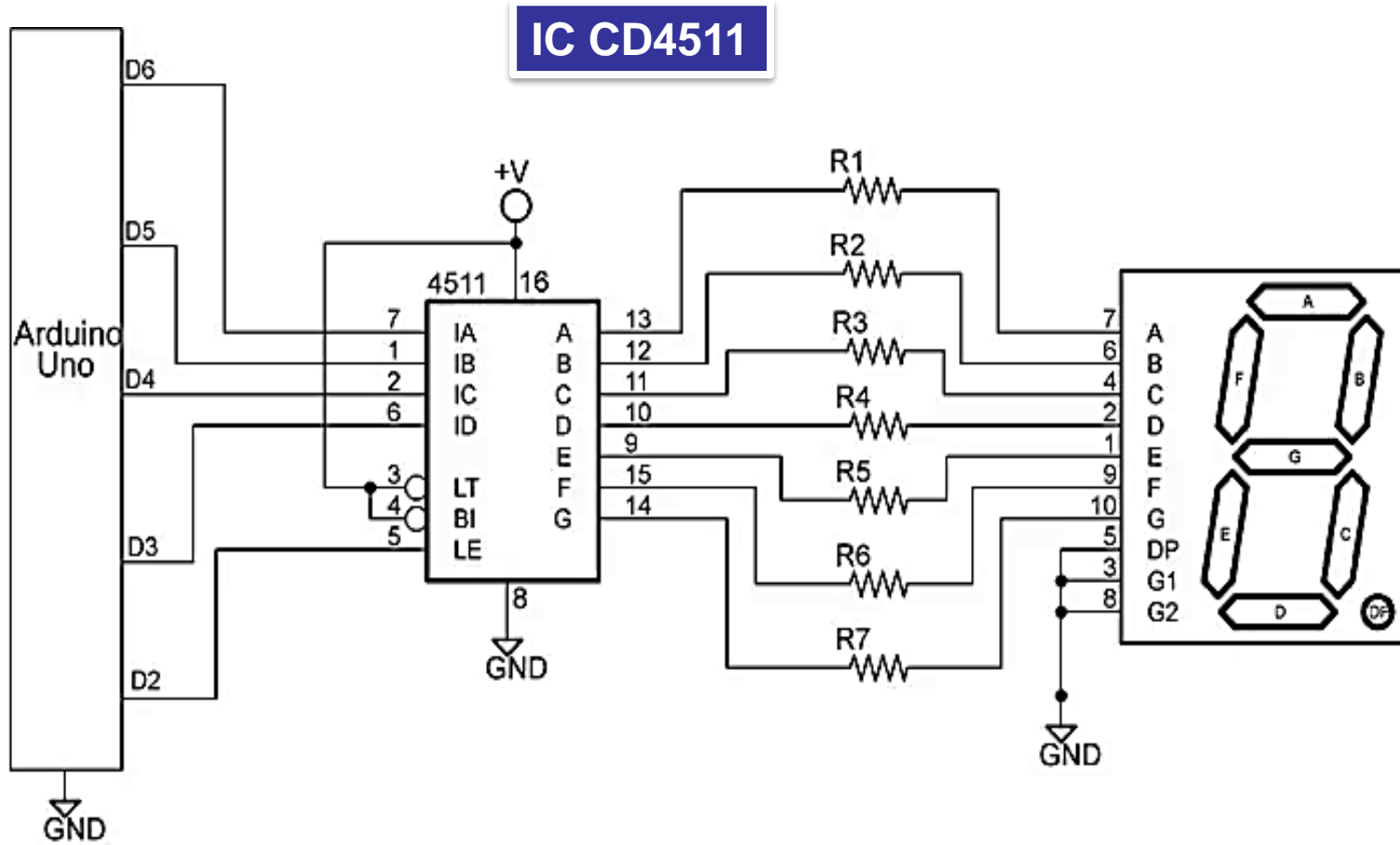


Driver Common Katoda

IC 7448

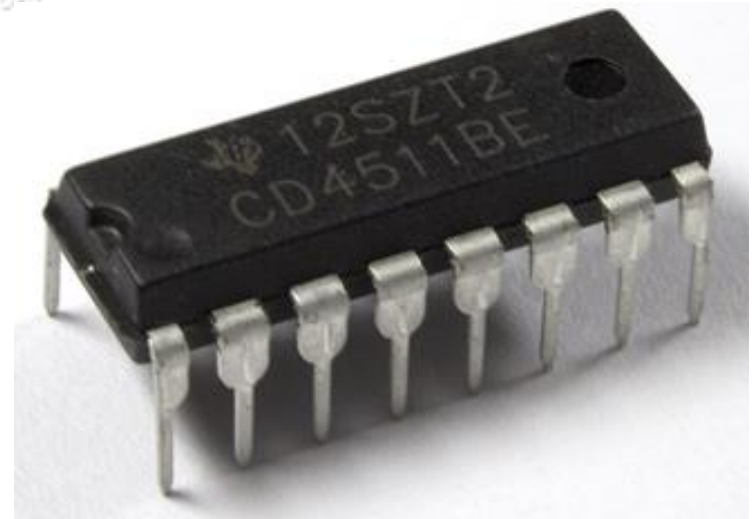
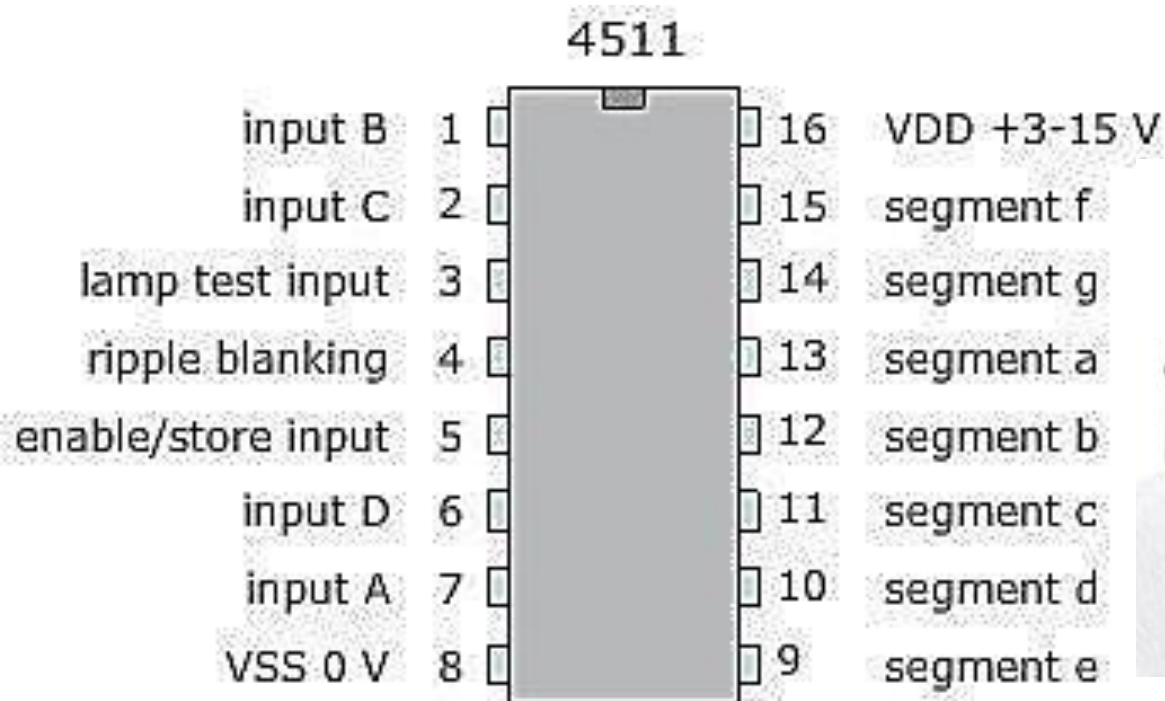


Driver Common Katoda



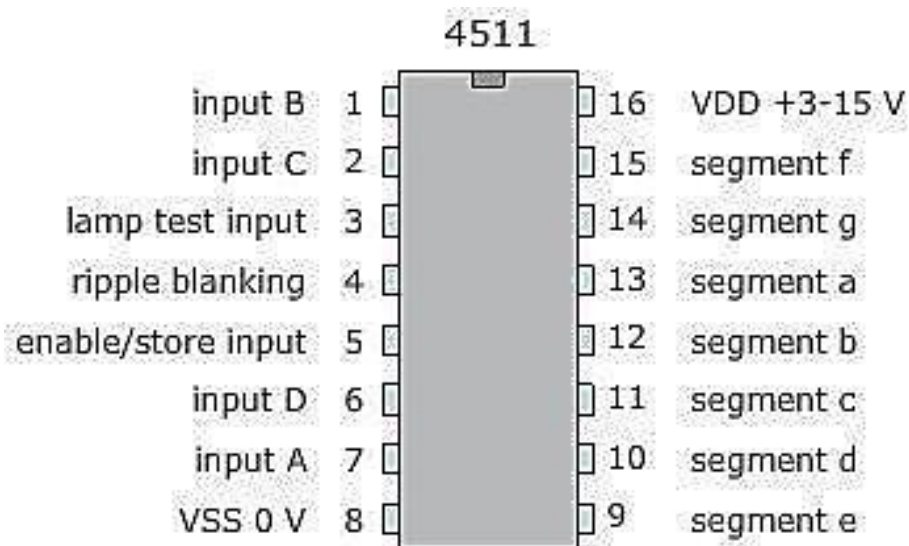
Driver Common Katoda

IC CD4511

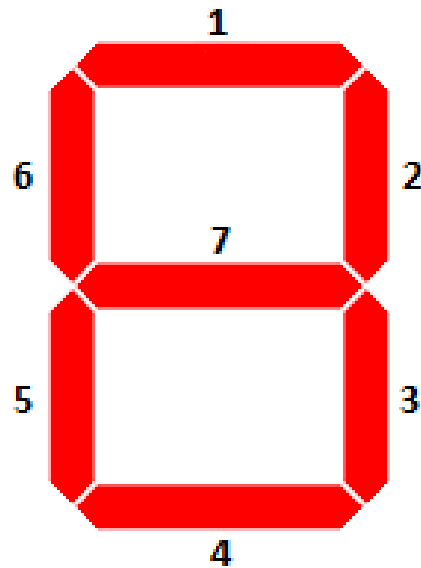


Tabel kebenaran IC CD4511

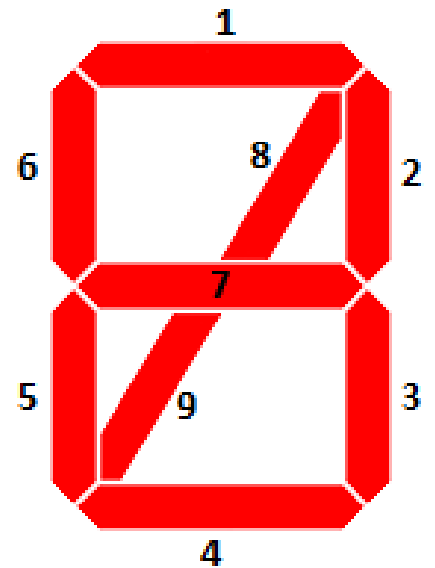
Decimal	Binary DCBA	7 Segment Code a b c d e f g
0	0000	1 1 1 1 1 1 0
1	0001	0 1 1 0 0 0 0
2	0010	1 1 0 1 1 0 1
3	0011	1 1 1 1 0 0 1
4	0100	0 1 1 0 0 1 1
5	0101	1 0 1 1 0 1 1
6	0110	0 0 1 1 1 1 1
7	0111	1 1 1 0 0 0 0
8	1000	1 1 1 1 1 1 1
9	1001	1 1 1 0 0 1 1



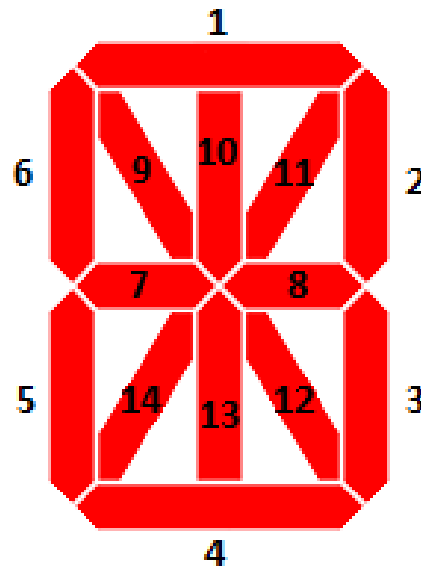
Segment Display



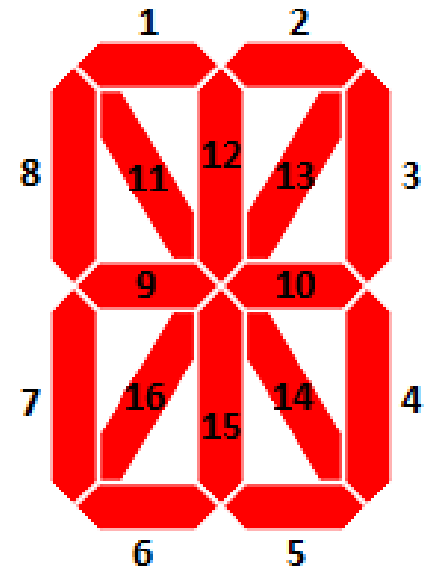
7-segment display



9-segment display

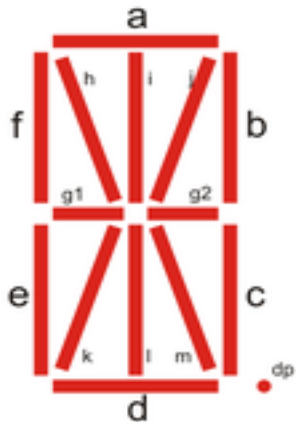


14-segment display



16-segment display

Fourteen-segment display



Decimal

Hexadecimal Encoding of Decimal Numbers for 14-segment display

Digit	0	1	2	3	4	5	6	7	8	9
Hex Code	0xC3F	0x406	0xDB	0x8F	0xE6	0xED	0xFD	0x1401	0xFF	0xE7

Latin alphabet

Hexadecimal Encoding of Latin Alphabet for 14-segment display

Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Hex code	0xF7	0x128F	0x39	0x120F	0xF9	0xF1	0xBD	0xF6	0x1209	0x1E	0x2470	0x38	0x536	0x2136

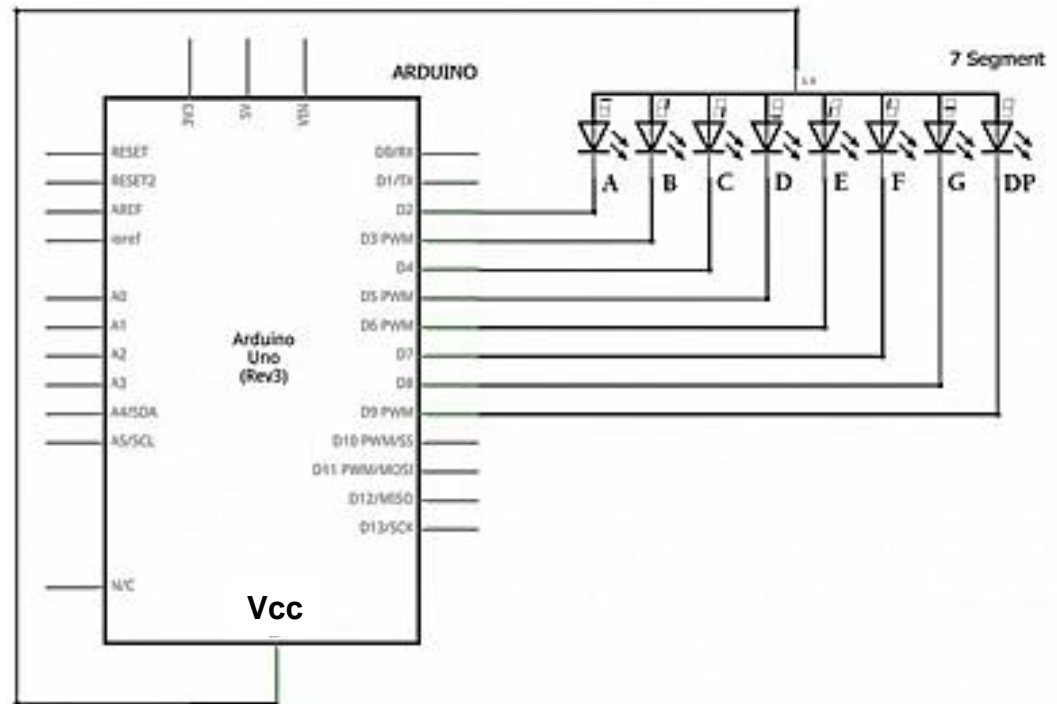
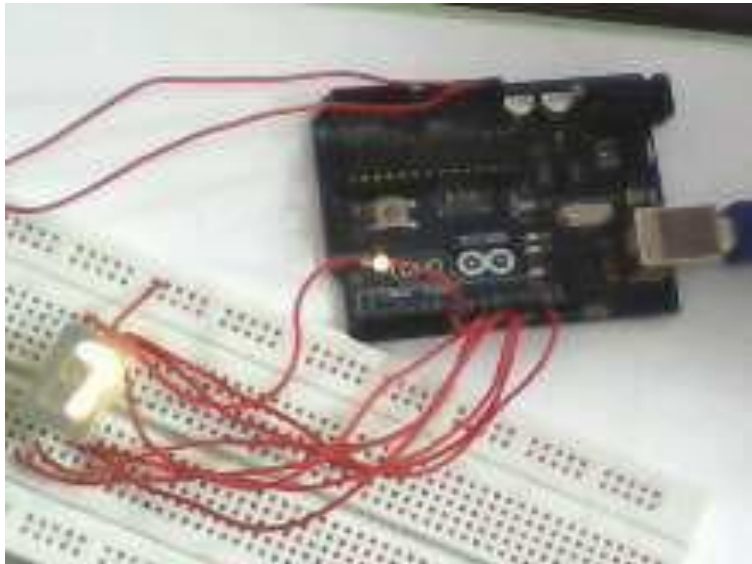
Alphabet	O	P	Q	R	S	T	U	V	W	X	Y	Z
Hex code	0x3F	0xF3	0x203F	0x20F3	0x18D	0x1201	0x3E	0xC30	0x2836	0x2D00	0x1500	0xC09

ALAT & RANGKAIAN

Hardware :

- Arduino Uno Board
- 1 buah Seven segmet common Anoda
- Kabel jumper

Rangkaian :

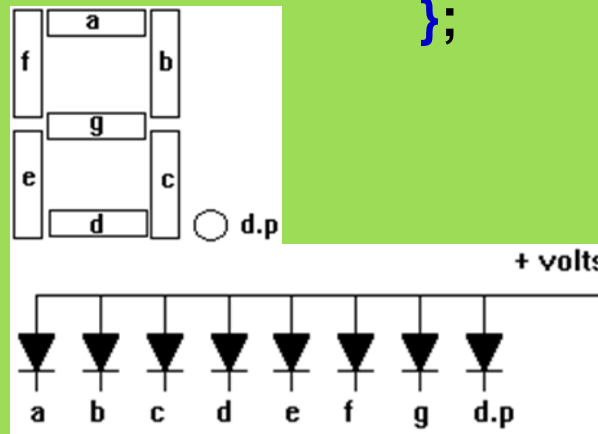


PROGRAM-1a: 7-S Common Anoda

```
int A=13, B=12, C=11, D=10;
int E=9, F=8, G=7, DP=6;
void setup() {
  pinMode(A, OUTPUT);
  pinMode(B, OUTPUT);
  pinMode(C, OUTPUT);
  pinMode(D, OUTPUT);
  pinMode(E, OUTPUT);
  pinMode(F, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(DP, OUTPUT);
}
```

```
void digit0 () {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
```

```
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, HIGH);
  digitalWrite(DP, HIGH);
};
```

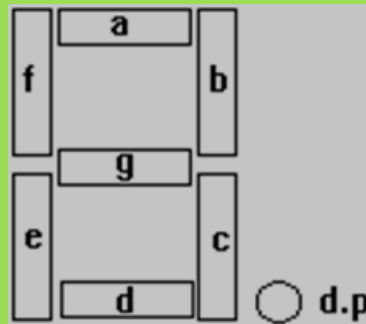


Pin 7-S	Pin Arduino
7 (a)	13
6 (b)	12
4 (c)	11
2 (d)	10
1 (e)	9
9 (f)	8
10 (g)	7
5 (DP)	6
3,8 (CA)	Vcc (+5V)

PROGRAM - lanjutan

```
void digit1 () {  
  digitalWrite(A,HIGH);  
  digitalWrite(B, LOW);  
  digitalWrite(C, LOW);  
  digitalWrite(D, HIGH);  
  digitalWrite(E, HIGH);  
  digitalWrite(F, HIGH);  
  digitalWrite(G, HIGH);  
  digitalWrite(DP, HIGH);  
};
```

```
void digit2 () {  
  digitalWrite(A,LOW);  
  digitalWrite(B, LOW);  
  digitalWrite(C, HIGH);  
  digitalWrite(D, LOW);  
  digitalWrite(E, LOW);  
  digitalWrite(F, HIGH);  
  digitalWrite(G, LOW);  
  digitalWrite(DP, HIGH);  
};
```



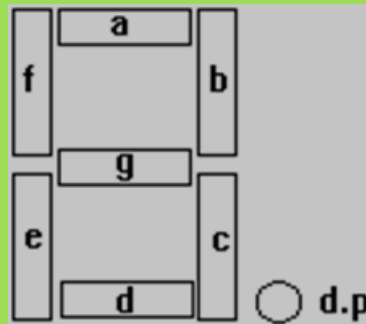
```
void digit3 () {  
  digitalWrite(A,LOW);  
  digitalWrite(B, LOW);  
  digitalWrite(C, LOW);  
  digitalWrite(D, LOW);  
  digitalWrite(E, HIGH);  
  digitalWrite(F, HIGH);  
  digitalWrite(G, LOW);  
  digitalWrite(DP, HIGH);  
};
```

```
void digit4 () {  
  digitalWrite(A,HIGH);  
  digitalWrite(B, LOW);  
  digitalWrite(C, LOW);  
  digitalWrite(D, HIGH);  
  digitalWrite(E, HIGH);  
  digitalWrite(F, LOW);  
  digitalWrite(G, LOW);  
  digitalWrite(DP, HIGH);  
};
```

PROGRAM - lanjutan

```
void digit5 () {  
  digitalWrite(A,LOW);  
  digitalWrite(B, HIGH);  
  digitalWrite(C, LOW);  
  digitalWrite(D, LOW);  
  digitalWrite(E, HIGH);  
  digitalWrite(F, LOW);  
  digitalWrite(G, LOW);  
  digitalWrite(DP, HIGH);  
};
```

```
void digit6 () {  
  digitalWrite(A,LOW);  
  digitalWrite(B, HIGH);  
  digitalWrite(C, LOW);  
  digitalWrite(D, LOW);  
  digitalWrite(E, LOW);  
  digitalWrite(F, LOW);  
  digitalWrite(G, LOW);  
  digitalWrite(DP, HIGH);  
};
```

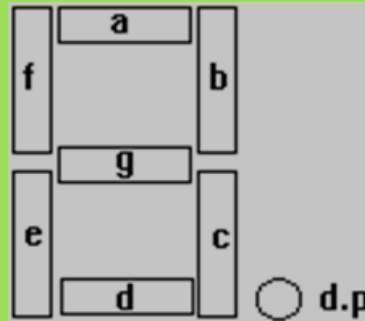


```
void digit7 () {  
  digitalWrite(A,LOW);  
  digitalWrite(B, LOW);  
  digitalWrite(C, LOW);  
  digitalWrite(D, HIGH);  
  digitalWrite(E, HIGH);  
  digitalWrite(F, HIGH);  
  digitalWrite(G, HIGH);  
  digitalWrite(DP, HIGH);  
};
```

```
void digit8 () {  
  digitalWrite(A, LOW);  
  digitalWrite(B, LOW);  
  digitalWrite(C, LOW);  
  digitalWrite(D, LOW);  
  digitalWrite(E, LOW);  
  digitalWrite(F, LOW);  
  digitalWrite(G, LOW);  
  digitalWrite(DP, HIGH);  
};
```


PROGRAM - lanjutan

```
void digit9 () {  
    digitalWrite(A, LOW);  
    digitalWrite(B, LOW);  
    digitalWrite(C, LOW);  
    digitalWrite(D, LOW);  
    digitalWrite(E, HIGH);  
    digitalWrite(F, LOW);  
    digitalWrite(G, LOW);  
    digitalWrite(DP, HIGH);  
};
```



```
void showdigit (int digit)  
{  
    switch (digit) {  
        case 0:  
            digit0 ();  
            break;  
        case 1:  
            digit1 ();  
            break;
```

```
        case 2:  
            digit2 ();  
            break;  
        case 3:  
            digit3 ();  
            break;  
        case 4:  
            digit4 ();  
            break;
```

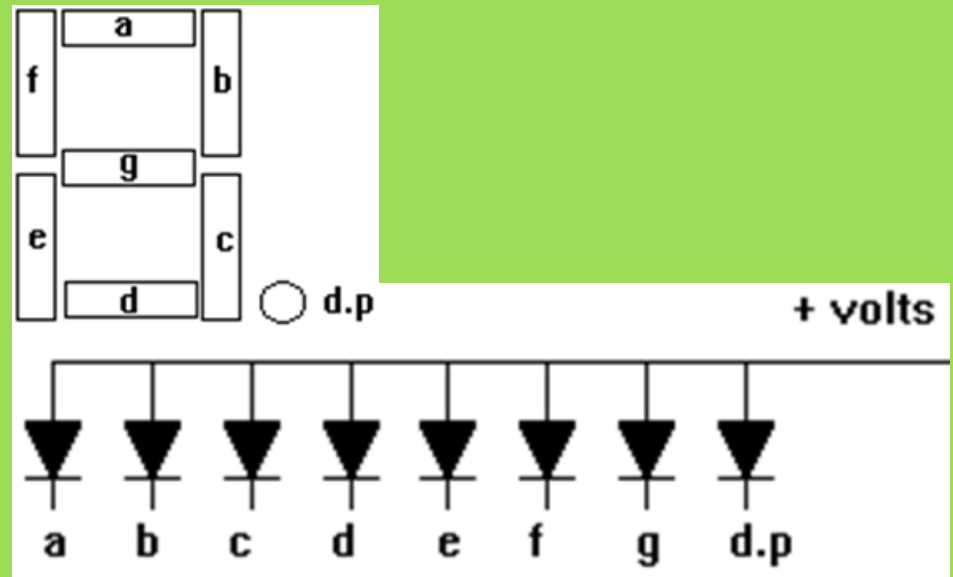
```
        case 5:  
            digit5 ();  
            break;  
        case 6:  
            digit6 ();  
            break;  
        case 7:  
            digit7 ();  
            break;  
        case 8:  
            digit8 ();  
            break;  
        case 9:  
            digit9 ();  
            break;  
        default:  
            break;  
    };  
};
```

PROGRAM - lanjutan

```
void loop() {  
  for (int i=0;i<10;i++) { //counting from 0 to 9  
    showdigit(i);  
    delay (1000); // 1000ms= 1s delay  
    if (i%2) { digitalWrite(DP, HIGH); }  
    else {digitalWrite(DP, LOW); };  
  };  
};
```

PROGRAM-1b: 7-S Common Anoda

```
byte seven_seg_digits[10][7] = {  
    { 0,0,0,0,0,0,1 }, // = 0  
    { 1,0,0,1,1,1,1 }, // = 1  
    { 0,0,1,0,0,1,0 }, // = 2  
    { 0,0,0,0,1,1,0 }, // = 3  
    { 1,0,0,1,1,0,0 }, // = 4  
    { 0,1,0,0,1,0,0 }, // = 5  
    { 0,1,0,0,0,0,0 }, // = 6  
    { 0,0,0,1,1,1,1 }, // = 7  
    { 0,0,0,0,0,0,0 }, // = 8  
    { 0,0,0,0,1,0,0 }, // = 9  
};
```



PROGRAM - lanjutan

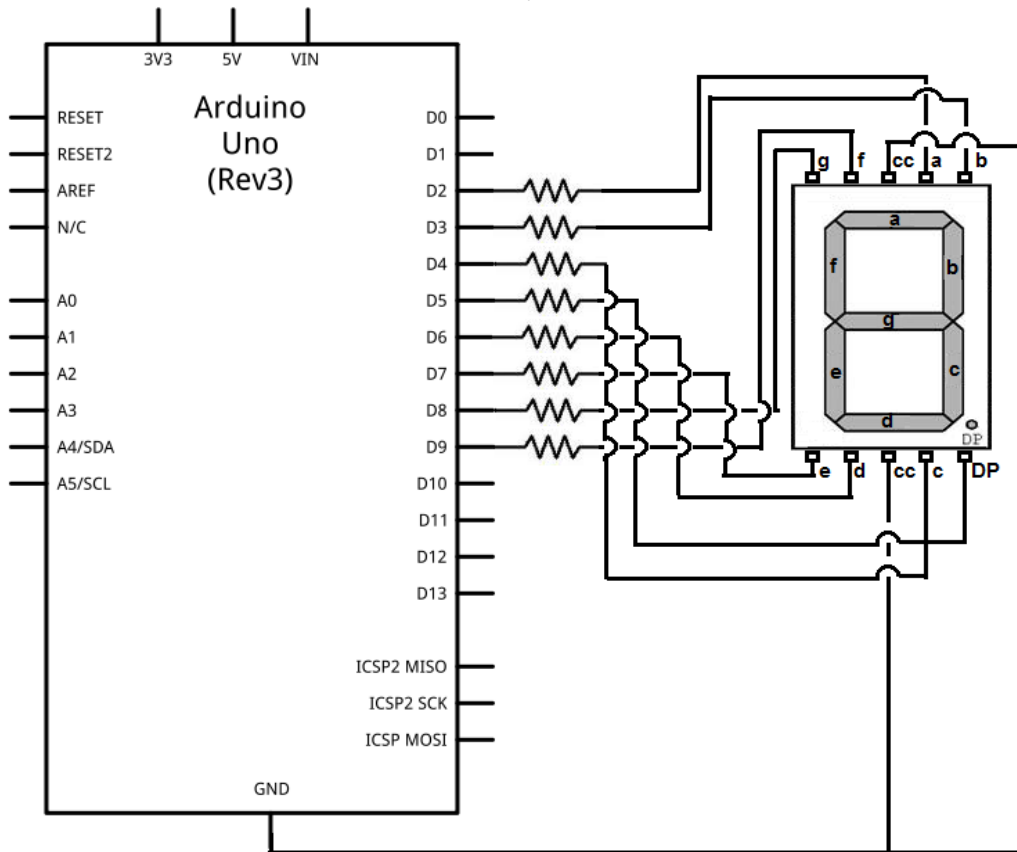
```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(12, OUTPUT);  
  pinMode(11, OUTPUT);  
  pinMode(10, OUTPUT);  
  pinMode(9, OUTPUT);  
  pinMode(8, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(6, OUTPUT);  
  writeDot(1); // padamkan tanda "dot" (titik)  
}  
void writeDot(byte dot) {  
  digitalWrite(6, dot);  
}  
void sevenSegWrite(byte digit) {  
  byte pin = 13;  
  for (byte segCount = 0; segCount < 7; ++segCount) {  
    digitalWrite(pin, seven_seg_digits[digit][segCount]);  
    ++pin;  
  }  
}
```

Pin 7-S	Pin Arduino
7 (a)	13
6 (b)	12
4 (c)	11
2 (d)	10
1 (e)	9
9 (f)	8
10 (g)	7
5 (DP)	6
3,8 (CA)	Vcc (+5V)

PROGRAM - *lanjutan*

```
void loop() {  
  
    for (byte count = 0; count < 10 ; ++count) {  
        sevenSegWrite(count);  
        delay(1000);  
    }  
  
    for (byte count = 9; count > 0; --count) {  
        sevenSegWrite(count-1);  
        delay(1000);  
    }  
}
```

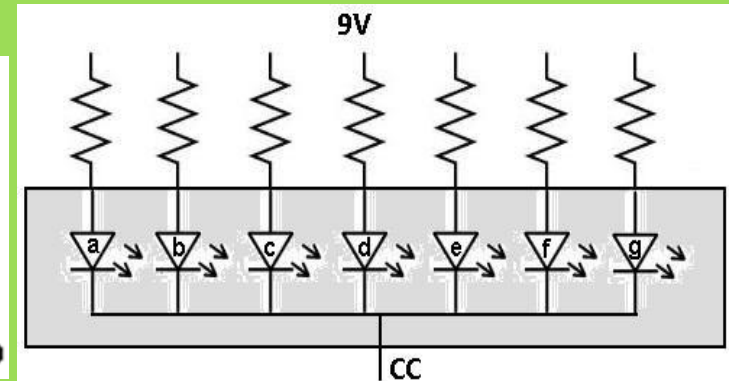
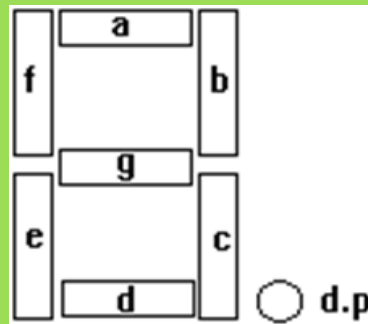
Skema Rangkaian Seven Segment Common Katoda



Pin 7-S	Pin Arduino
7 (a)	2
6 (b)	3
4 (c)	4
2 (d)	6
1 (e)	7
9 (f)	9
10 (g)	8
5 (DP)	5
3,8 (CC)	GND

PROGRAM-2: 7-S Common Katoda

```
const byte numeral[11]= {  
  B11111100, //0  
  B01100000, //1  
  B11011010, //2  
  B111110010, //3  
  B01100110, //4  
  B10110110, //5  
  B00111110, //6  
  B11100000, //7  
  B11111110, //8  
  B11100110, //9  
  B00000000, //off  
};
```



PROGRAM - *lanjutan*

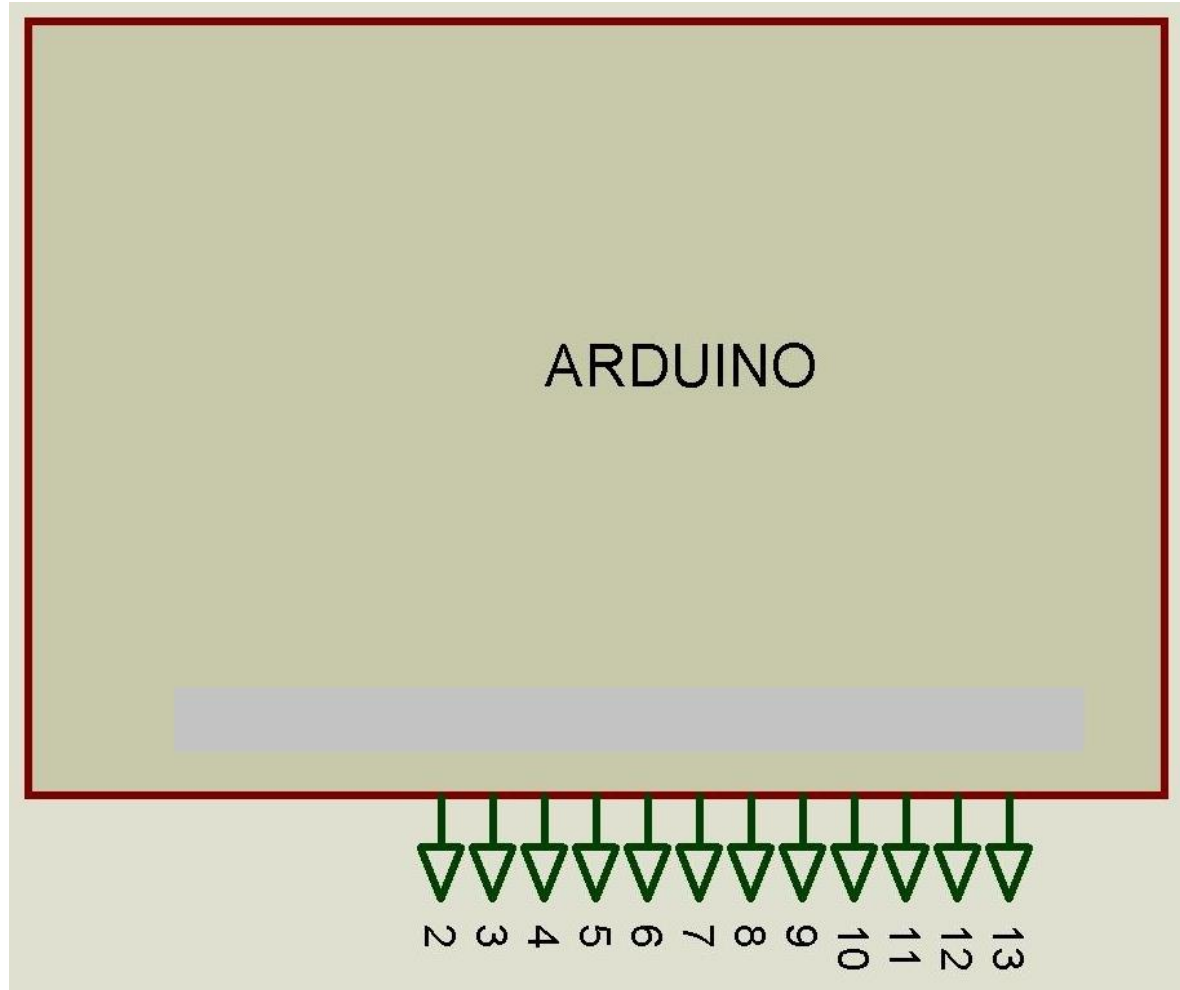
```
void setup()
{
  for (int i=0; i < 8; i++)
  {
    pinMode(segmentPins[i], OUTPUT);
  }
}
void loop()
{
  for (int i=0; i <=10; i++)
  {
    showDigit(i);
    delay(1000);
  }
  delay(2000);
}
```


PROGRAM - *lanjutan*

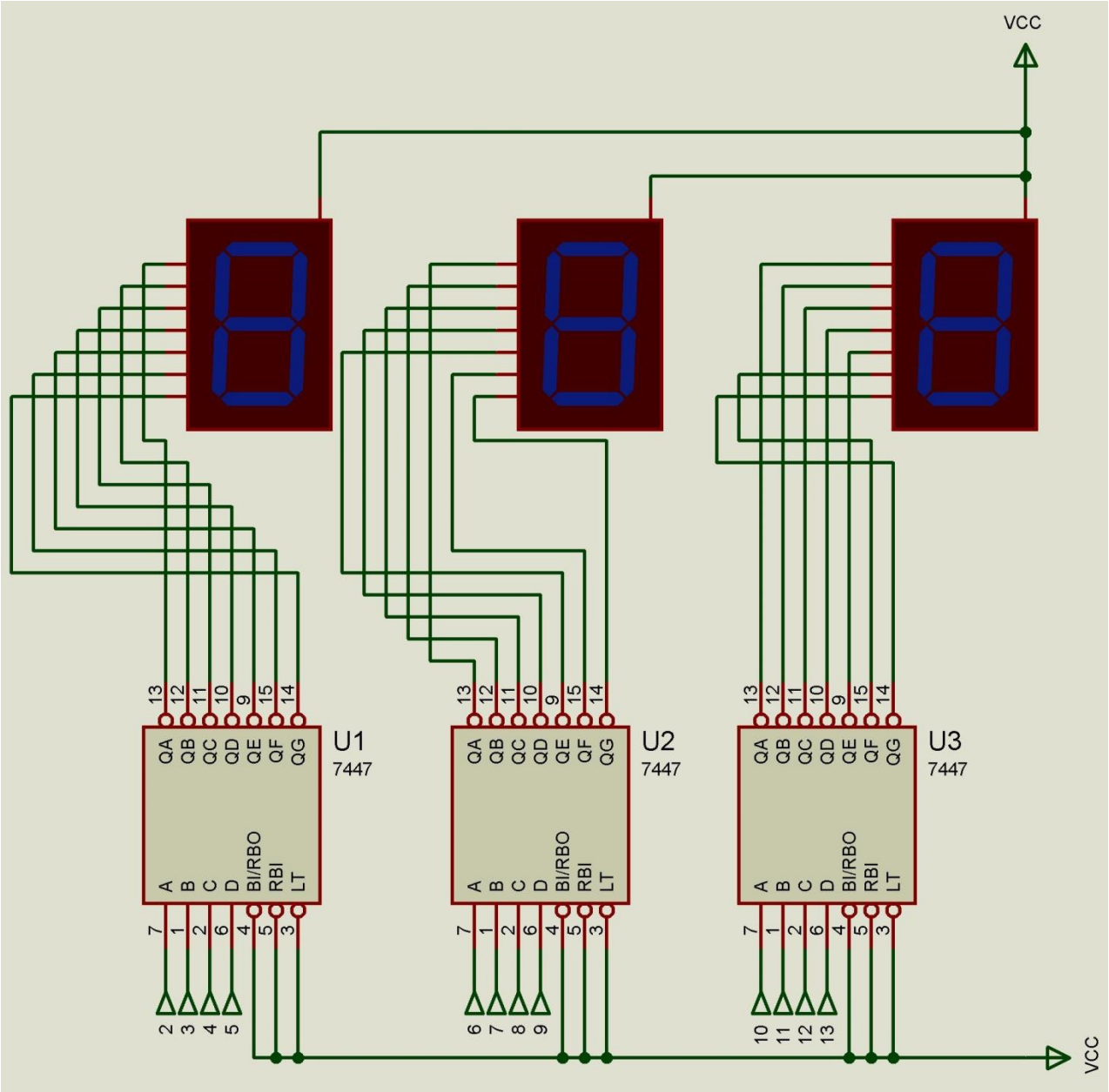
```
void showDigit (int number)
{
  boolean isBitSet;

  for (int segment=1; segment < 8; segment++)
  {
    IsBitSet= bitRead(numeral[number], segment);
    digitalWrite(segmentPins[segment], isBitSet);
  }
}
```

Rangkaian Antarmuka (Tiga 7s):



Rangkaian Antarmuka (Tiga 7S):



Hasil :

- Catat hasil nyala LED pada 7 Segment dan foto untuk dokumentasi.
- Buat laporan hasil dari percobaan

Latihan :

1. Buatlah eksperimen dengan menggunakan 1 driver (IC TTL 74LS47) dan sebuah 7S Common Anoda.
2. Buatlah eksperimen dengan menggunakan 1 driver (IC TTL 74LS48 atau CD4511) dan sebuah 7S Common Katoda.
3. Buat aplikasi kontrol 7S Common Anoda dan driver 74LS47 dengan menggunakan saklar sebagai input untuk Set data (Up/down-counter) dan RESET data.

Tugas-1

1. Buat Program dari rangkaian antarmuka mikrokontroler dengan dua 7S Common Katoda tanpa menggunakan driver
2. Buat Rangkaian dan Program antarmuka mikrokontroler dengan dua 7S Common Anoda tanpa menggunakan driver
3. Buat Program dari rangkaian antarmuka mikrokontroler dengan tiga 7S Common Anoda menggunakan driver 74LS47