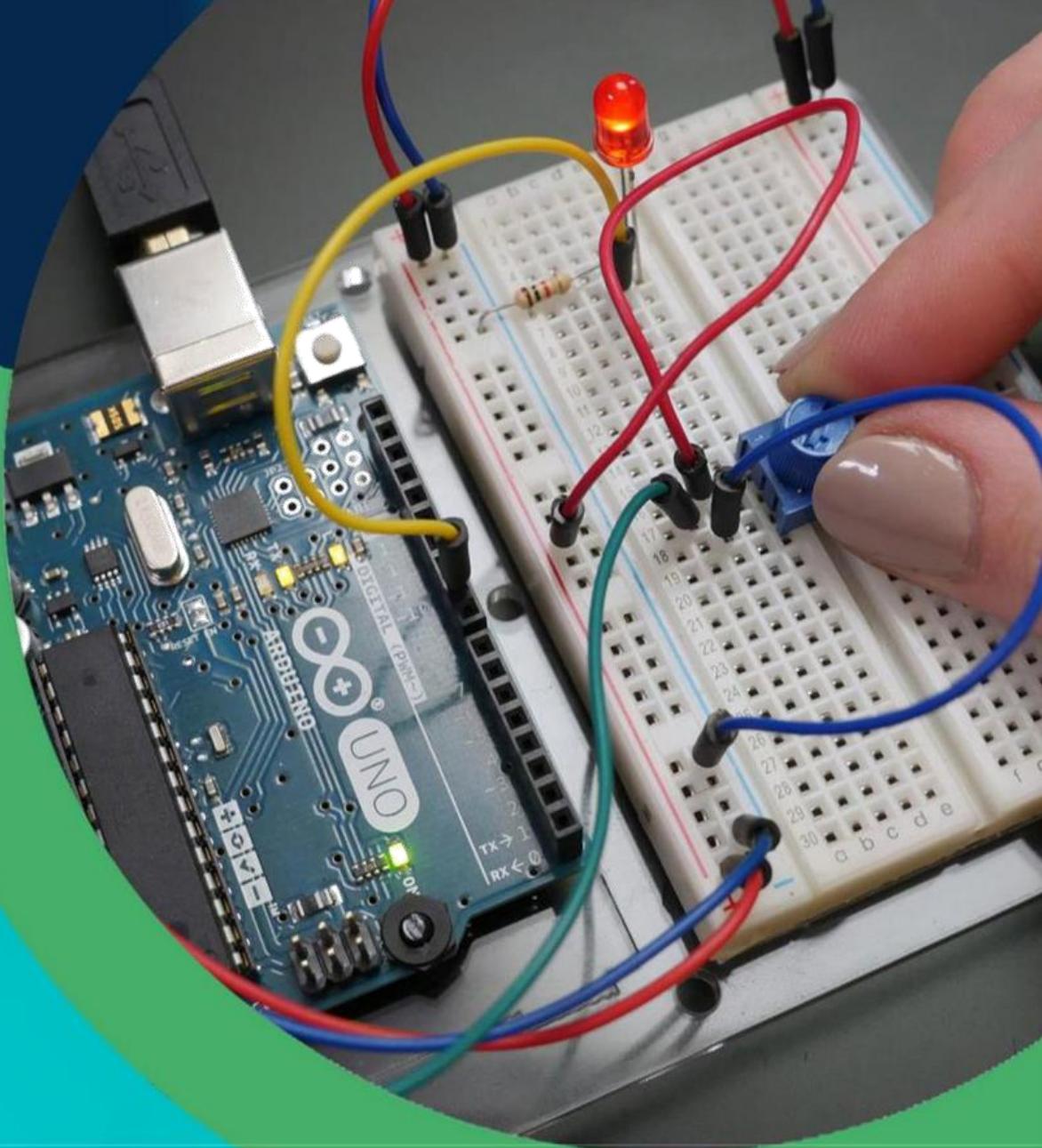


2017



# BUKU PRAKTIKUM **MIKROPROSESOR & ANTARMUKA 1**

Disusun Oleh :  
Akuwan Saleh, S.ST.,MT.

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

## KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Panyayang, Kami panjatkan puja dan puji syukur atas kehadirat-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan buku mikroprosesor dan antarmuka 1.

Buku ini telah kami susun dengan maksimal dan mendapatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan buku ini. Untuk itu kami menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam pembuatan buku ini.

Terlepas dari semua itu, Kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar kami dapat memperbaiki buku ini.

Akhir kata kami berharap buku mikroprosesor dan antarmuka 1 ini dapat memberikan manfaat maupun inspirasi terhadap pembaca.

Surabaya, 28 Juli 2017

Penyusun

## DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
PENDAHULUAN.....	1
ANTARMUKA MIKROKONTROLER DENGAN LIGHT EMITTING DIODE (LED).....	5
ANTARMUKA MIKROKONTROLER DENGAN SAKLAR.....	12
LED DOT MATRIX.....	17
KONTROL NYALA LED DENGAN KEYPAD.....	23
KONTROL DISPLAY 7- SEGMENT.....	29
PEMROGRAMAN LCD.....	38
PEMROGRAMAN MELODY.....	47
ANALOG INPUT (ADC).....	52
KOMUNIKASI SERIAL.....	60
APLIKASI LM35.....	65
ANTARMUKA MIKROKONTROLER DENGAN LDR.....	70
ANTARMUKA MIKROKONTROLER DENGAN LAMPU AC 220V.....	77
ANTARMUKA MIKROKONTROLER DENGAN MOTOR DC.....	83
ANTARMUKA MIKROKONTROLER DENGAN ULTRASONIC.....	88
DAFTAR PUSTAKA.....	95

## PENDAHULUAN

### MIKROPROSESOR

#### A. PENGANTAR MIKROPROSESOR

Mikroprosesor adalah bagian CPU dari sebuah komputer, tanpa *Memory*, I/O, dan peripheral lain. Supaya dapat bekerja, mikroprosesor memerlukan perangkat pendukung seperti RAM, ROM dan I/O.

Mikrokomputer merupakan kombinasi mikroprosesor dengan I/O dan *Memory* (RAM/ROM yang dibuat dalam bentuk *Single Chip* yaitu *Single Chip Microcomputer* (SCM)).

#### B. SEJARAH MIKROPROSESOR

##### Alat penghitung

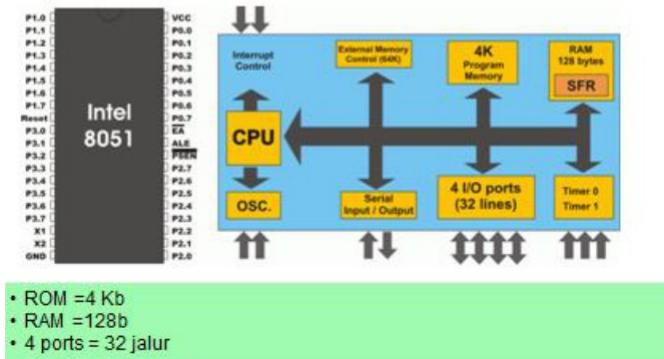
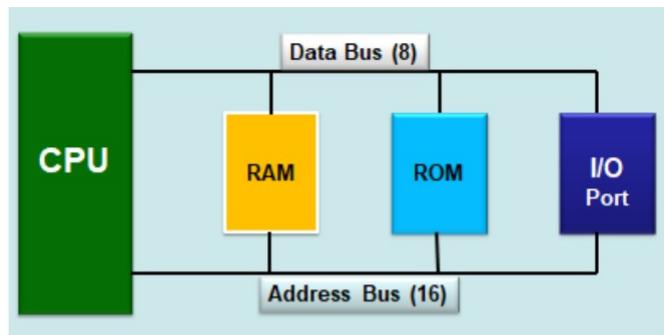
- Menghitung biji jagung dengan menggunakan jari (digit)
- *Sempoa*
- Tahun 1642 Blaise Pascal : mesin kalkulasi digital pertama kali
- Tahun 1671, Gottfried Wilhelm von Leibniz mesin hitung untuk operasi perkalian.
- Tahun 1820, Thomas Colmar (Charles Xavier Thomas) kalkulator mekanik mampu tambah, kurang, kali, dan bagi.
- Tahun 1812, Charles Babbage, kalkulator otomatis menjadikan ciri dari komputer elektronik masa kini. tahun 1822 membangun model aktif kecil untuk demonstrasi.
- 1941 kalkulator untuk membangkitkan mathematical table. Kalkulator elektronik pertama adalah ENIAC (for Electrical Numerical Integrator And Calculator), menggunakan 18000 tabung ruang hampa, 1800 kaki luas lantai bujur sangkar, dengan power 180 kW.
- Tabung ruang hampa segera digantikan dengan transistor pada tahun 1955, yang pada gilirannya digantikan oleh Integrated Circuit (IC) di tahun 1964.
- Tahun 1971 Central Processing Unit (CPU) – sekarang

## C. PERBEDAAN MIKROPROSESOR DAN MIKROKONTROLLER

### 1. Mikroprosesor

Mikroprosesor adalah sebuah single chip yang berisi CPU (Central Processing Unit). Untuk membentuk sebuah minimum sistem mikrokomputer dibutuhkan peralatan pendukung:

- RAM ( Random Acces Memory )
- ROM ( Read Only Memory )
- I/O ( Unit Input / Output )



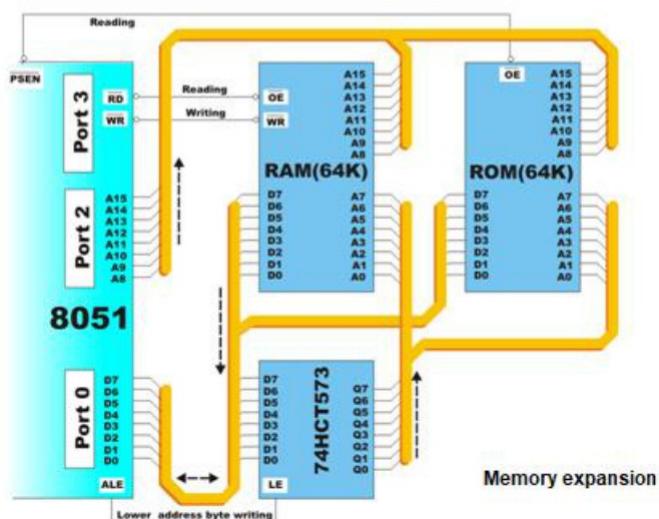
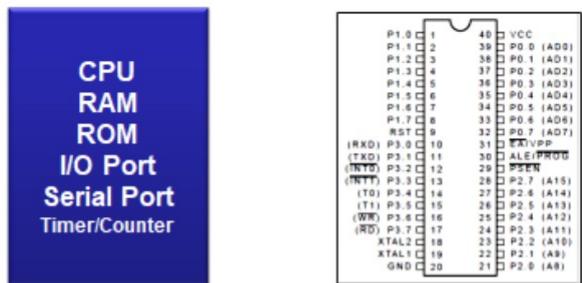
Gambar 1. Blok Diagram Mikroprosesor

### 2. Mikrokontroller

Mikrokontroller adalah sebuah single chip yang di dalamnya sudah berisi :

- CPU ( Central Processing Unit )
- RAM ( Random Acces Memory )

- ROM (Read Only Memory)
  - I/O (Unit Input / Output)
  - Timer/Counter , Serial COM Port.
  - Penggunaan perangkat I/O dan media



Gambar 2. Blok Diagram Mikrokontroller

Sebuah mikrokontroler tidak dapat bekerja bila tidak diberikan program kepadanya. Sistem kerja mikrokontroler dapat dirubah setiap saat sesuai dengan program yang diberikan kepadanya. Instruksi-instruksi perangkat lunak berbeda untuk masing-masing jenis mikrokontroller. Sebuah mikrokontroler tidak dapat memahami instruksi-instruksi yang berlaku pada mikrokontroler jenis lain. Penggunaan perangkat I/O dan media penyimpan program pada Mikrokontroller ialah menggunakan EEPROM. Keuntungan mikrokontroler

dibandingkan dengan mikroprosesor ialah mikrokontroler mempunyai memori (RAM/ROM) dan peralatan I/O.

Contoh:

Mikrokontroler buatan Intel dengan mikrokontroler buatan Zilog memiliki perangkat instruksi yang berbeda, seperti contoh dibawah ini :

*LD A,E*       $\Rightarrow$  *Z80*

*MOV AL,DL*       $\Rightarrow$  *Intel 80x86/88*

*MOV R0,A*       $\Rightarrow$  *MCS-51*

Instruksi-instruksi pada mikrokontroler dikenal sebagai bahasa pemrograman sistem mikrokontroler.

## PERCOBAAN 1

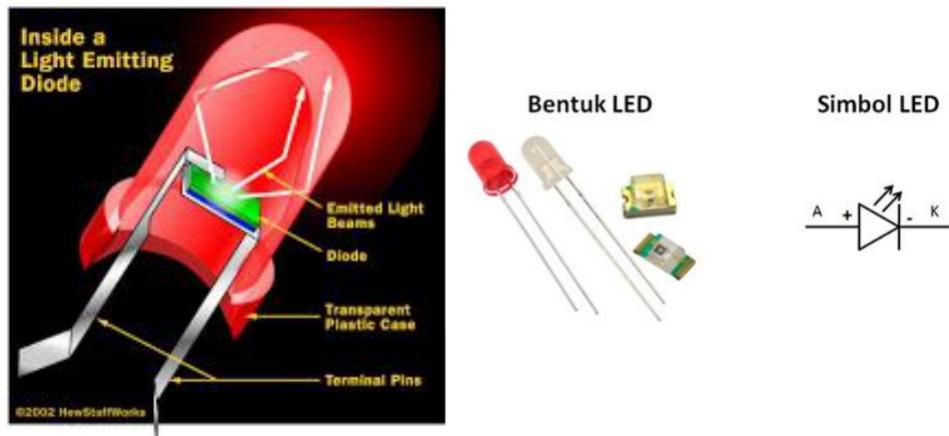
### ANTARMUKA MIKROKONTROLER DENGAN LIGHT EMITTING DIODE (LED)

#### A. TUJUAN

1. Dapat membuat proses kontrol display LED sebagai output.
2. Mengetahui dan memahami bagaimana memrogram mikrokontroler untuk menyalakan led.

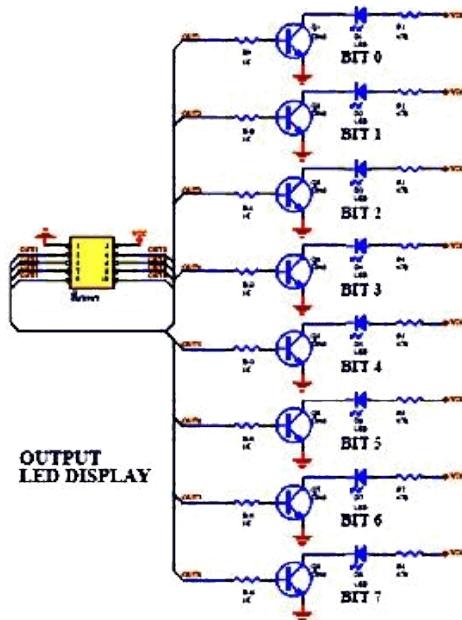
#### B. DASAR TEORI

LED (Light Emitting Diode) adalah sebuah sumber cahaya yang terbuat dari semikonduktor. LED digunakan sebagai lampu indikator dalam beberapa peralatan, banyak digunakan sebagai penerangan/lampu. Pada papan Arduino Uno terdapat sebuah LED kecil yang terhubung ke pin digital 13. LED ini dapat digunakan sebagai output saat user membuat sebuah program dan membutuhkan sebuah penanda dari jalannya program tersebut. Untuk menguji board arduino user dapat membuat program menyalakan LED pin-13 berkedip-kedip dalam jeda satu detik dengan perintah delay(1000).



Gambar 1. Bentuk LED dan Simbol LED

Untuk menyalakan sebuah LED perlu rangkaian tambahan dalam gambar rangkaian yang berupa sebuah transistor yang difungsikan sebagai saklar dan dua buah resistor untuk pembatas arus.



Gambar 2. Transistor yang difungsikan sebagai saklar dan dua buah resistor pembatas arus untuk menyalaikan LED.

Dalam modul I/O yang dipakai dalam praktikum, rangkaian LED tersebut dihubungkan ke sebuah pin pada board arduino. Contoh seperti pada gambar dibawah ini :

### Penulisan program dengan bahasa C

#### 1. Fungsi Setup

```
void setup(){
  / nothing to setup
{
```

Fungsi Setup diatas tidak melakukan setting apapun dan hanya diperlukan ketika terdapat kesalahan proses saat verifikasi atau upload sketsa.

## 2. Fungsi loop

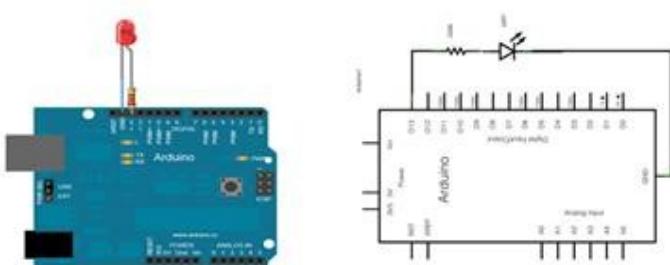
```
void loop()
{
    // Loop u/ LED berkedip.
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

Fungsi loop digunakan untuk memproses atau mengakses program secara berulang-ulang sesuai dengan kondisi yang diberikan.

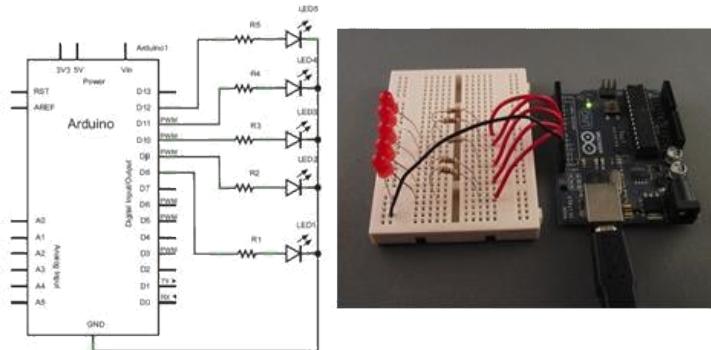
## C. HARDWARE

- Arduino Uno
  - Breadboard
  - Kabel Jumper
  - LED 8 buah
  - Resistor 220 ohm 8 buah (bila diperlukan)
  - Kabel Data USB

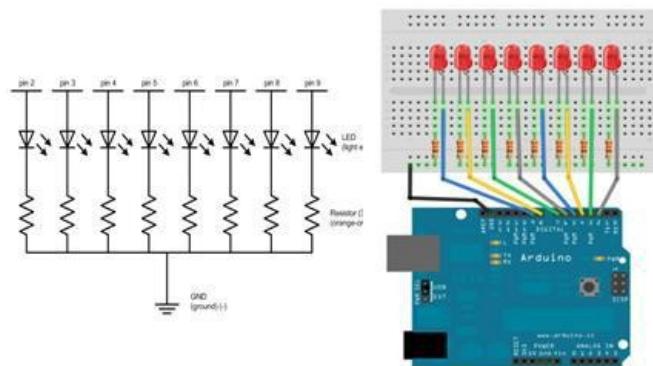
## D. RANGKAIAN



Gambar 3. Rangkaian Arduino dan 1-LED



Gambar 4. Rangkaian Arduino dan 5-LED



Gambar 5. Rangkaian Arduino dan 8-LED

## E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke komputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



4. Jalankan arduino dan buat programnya

## F. PROGRAM

1. led 13 berkedip dengan delay 1 sekon

```
int led = 13;

void setup() {
    // put your setup code here, to run once:
    pinMode(led, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(led, HIGH);
    delay(500);
    digitalWrite(led, LOW);
    delay(500);
}
```

2. Led 9 menyala dari redup ke terang dan sebaliknya

```
int led = 9;
int brightness = 0;
int fadeAmount = 5;

void setup() {
    // put your setup code here, to run once:
    pinMode(led, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    analogWrite(led, brightness);
    brightness = brightness + fadeAmount;
    if (brightness == 0 || brightness == 255) {
        fadeAmount = -fadeAmount ;
    }
    delay(30);
}
```

### 3. 8 led menyala bergantian dari kiri ke kanan lalu mati dari kanan ke kiri

```
//LED Pin Variables
int ledPins[] = {2,3,4,5,6,7,8,9};
void setup()
{
    for(int i = 0; i < 8; i++){
        pinMode(ledPins[i],OUTPUT);
    }
}
void loop()
{
    oneAfterAnotherLoop();
}
void oneAfterAnotherLoop() {
int delayTime = 100;
for(int i = 0; i <= 7; i++) {
digitalWrite(ledPins[i], HIGH);
delay(delayTime);
}
for(int i = 7; i >= 0; i--) {
digitalWrite(ledPins[i], LOW);
delay(delayTime);
}
}
void oneOnAtATime() {
int delayTime = 100;
for(int i = 0; i <= 7; i++) {
int offLED = i - 1;
if(i == 0) {
offLED = 7;
}
digitalWrite(ledPins[i], HIGH);
digitalWrite(ledPins[offLED], LOW);
delay(delayTime);
}
}
}
void inAndOut() {
int delayTime = 100;
for(int i = 0; i <= 3; i++) {
int offLED = i - 1;
if(i == 0) {
offLED = 3;
}
int onLED1 = 3 - i;
int onLED2 = 4 + i;
int offLED1 = 3 - offLED;
int offLED2 = 4 + offLED;
digitalWrite(ledPins[onLED1], HIGH);
digitalWrite(ledPins[onLED2], HIGH);
digitalWrite(ledPins[offLED1], LOW);
digitalWrite(ledPins[offLED2], LOW);
delay(delayTime);
}
for(int i = 3; i >= 0; i--) {
int offLED = i + 1;
if(i == 3) {
offLED = 0;
}
int onLED1 = 3 - i;
int onLED2 = 4 + i;
int offLED1 = 3 - offLED;
int offLED2 = 4 + offLED;
digitalWrite(ledPins[onLED1], HIGH);
digitalWrite(ledPins[onLED2], HIGH);
digitalWrite(ledPins[offLED1], LOW);
digitalWrite(ledPins[offLED2], LOW);
delay(delayTime);
}
}
```

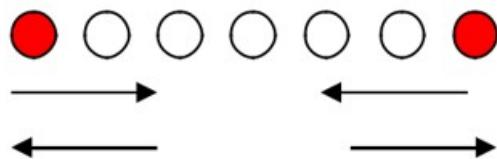
### 4. 5 led menyala bergantian dari kanan ke kiri

```
int ledArray[]={8,9,10,11,12};
int count = 0;
int timer=75;
void setup() {
    // put your setup code here, to run once:
for(count=0;count<5;count++)
{
    pinMode(ledArray[count],OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
for(count=0;count<5;count++)
{
    digitalWrite(ledArray[count],HIGH);
    delay(timer);
    digitalWrite(ledArray[count],LOW);
    delay(timer);
}
}
```

## G. LATIHAN

1. Buatlah eksperimen untuk mengontrol nyala LED-1 di pin 13 dan LED-2 di pin 9
  - Jika LED-1 nyala, LED-2 padam
  - Jika LED-1 padam, LED-2 nyala.LED-1 dan LED-2 nyala dan padam bergantian secara berulang-ulang.
2. Buatlah program untuk mengontrol nyala 5 LED bergerak dari LED-1 ke LED-5 dan sebaliknya dari LED-5 ke LED-1 berulang 3x kemudian 5 LED berkedip 2x.
3. Buat program untuk menyalakan led dengan tampilan :



4. Buatlah program untuk menyalakan LED 0 sampai LED 7 melompat 2, nyala LED dimulai dari LED 0, kemudian LED 2, 4, 6 kembali ke LED 0 berulang terus menerus.

## PERCOBAAN 2

### ANTARMUKA MIKROKONTROLER DENGAN SAKLAR

#### A. TUJUAN

1. Mampu membuat rangkaian antarmuka mikrokontroler dengan rangkaian input saklar.
2. Dapat membuat program bahasa C pada mikrokontroler untuk membaca data input dari saklar.

#### B. DASAR TEORI

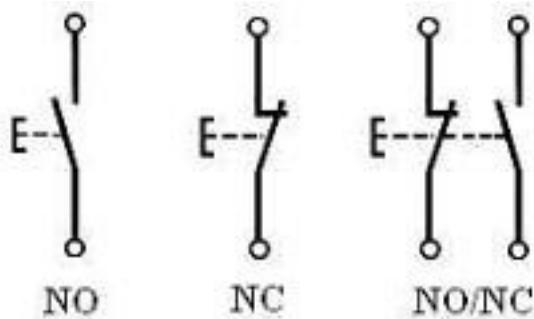
Saklar toggle adalah salah satu saklar elektrik yang digerakkan secara manual oleh batang mekanik. Saklar toggle tersedia dalam berbagai bentuk dan ukuran, serta digunakan dalam berbagai aplikasi. Dalam modul I/O, diberi tambahan berupa rangkaian LED untuk melihat secara langsung logika yang ada pada saklar.

Membuat Arduino menerima masukan (input) menggunakan tombol (push buttons) yang sederhana. Koneksi untuk penggunaan tombol juga sangat mudah. Ada tambahan komponen yaitu resistor, yang bias berfungsi sebagai *pull-up* atau *pull-down*. Kenapa diperlukan resistor? Karena Arduino akan 'membaca' tombol ditekan atau tidak dari arus yang melewatinya, apakah HIGH atau LOW.

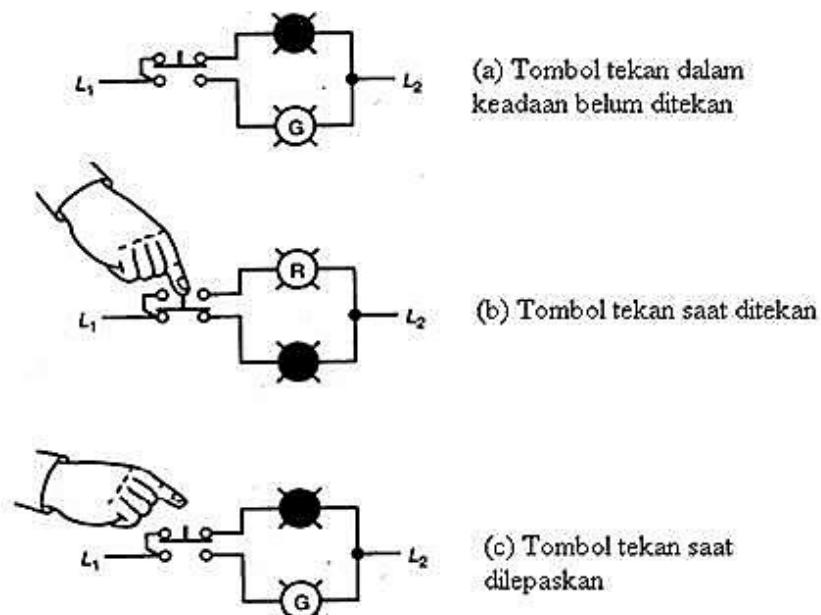
#### Tombol Tekan (Push Button)

- Tombol tekan NO (Normally Open) menyambung rangkaian ketika tombol ditekan dan kembali pada posisi terputus ketika tombol dilepas.
- Tombol tekan NC (Normally Closed) akan memutus rangkaian apabila tombol ditekan dan kembali pada posisi terhubung ketika tombol dilepaskan.
- Tombol tekan yang memiliki fungsi ganda, yakni sudah dilengkapi oleh dua jenis kontak, baik NO maupun NC. Jadi tombol tekan tersebut dapat difungsikan sebagai NO, NC atau keduanya. Ketika tombol ditekan,

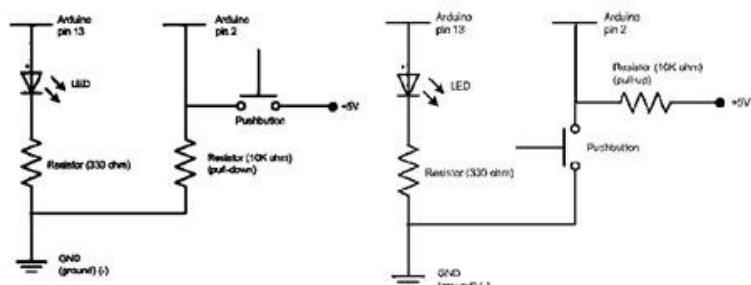
terdapat kontak yang terputus (NC) dan ada juga kontak yang terhubung (NO).



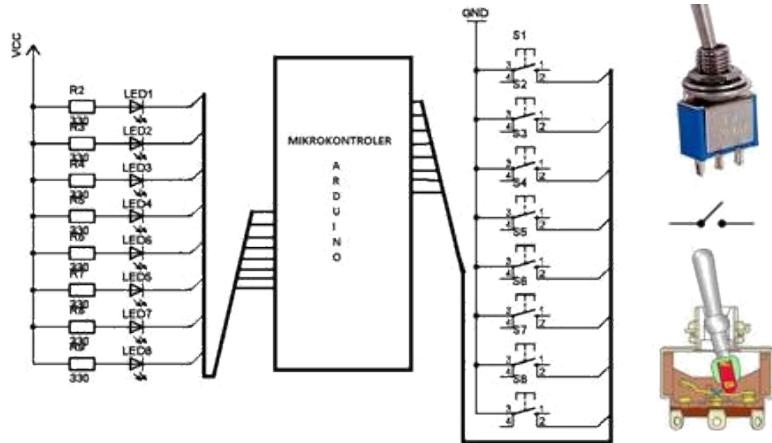
Gambar 1. Skema Rangkaian Push Button NO, NC, Dan NO/NC



Gambar 2. Prinsip Kerja Push Button



Gambar 3. Skema Rangkaian Push Button dengan Pull-Down Resistor (Kiri) dan dengan Pull-Up Resistor (Kanan).



Gambar 4. Skema Rangkaian Saklar Toggle

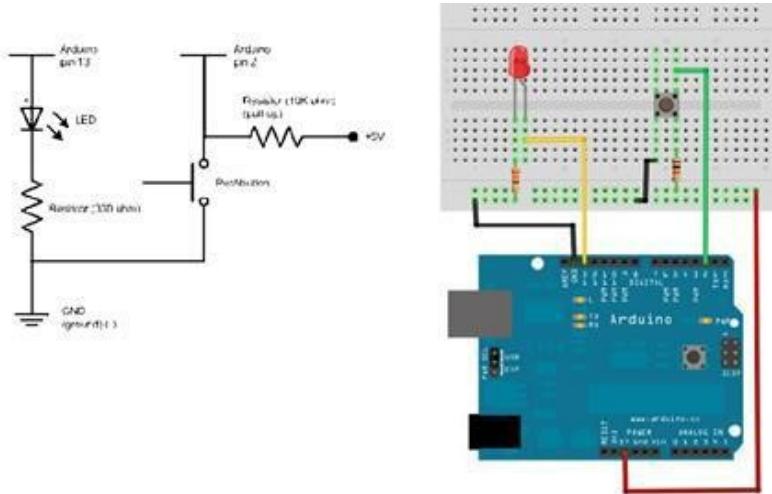
### Penulisan program dengan bahasa C

```

const int buttonPin = 2; //button Pin untuk variabel pin2 arduino
const int ledPin = 13; //ledPin untuk variabel pin13 arduino
int button State = 0;
void setup()
{
    pinMode(ledPin, OUTPUT); //ledPin sebagai output
    pinMode(buttonPin, INPUT); //buttonPin sebagai
    input }
void loop()
{
    buttonState = digitalRead(buttonPin);
    //Jika saklar on atau buttonState = high, tulis data logika high ke ledPin
    if (buttonState == HIGH) {
        digitalWrite(ledPin, HIGH);
    }
    //Jika saklar off atau buttonState = low, tulis data logika low ke
    ledPin else {
        digitalWrite(ledPin, LOW);
    }
}

```

## C. RANGKAIAN



Gambar 5. Skema Rangkaian Push Button Dengan Pull-Up Resistor Dan Arduino

## D. LANGKAH PERCOBAAN

1. Tancapkan usb serial kekemputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



4. Jalankan arduino dan buat programnya

## E. PROGRAM

### Program-1

```
const int buttonPin = 2;
const int ledPin = 13;
int buttonState = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == LOW) {
```

```

        digitalWrite(ledPin, LOW);
    }
else {
    digitalWrite(ledPin, HIGH);
}
}

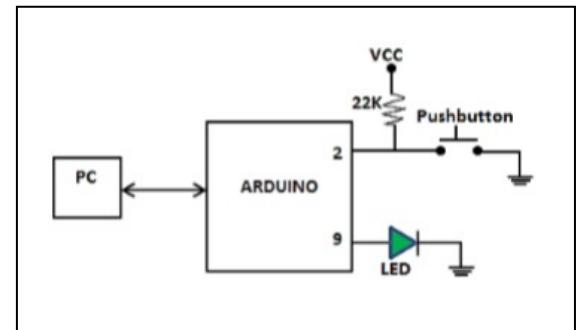
```

Hasil :

Button di pin arduino 2 dan led di pin arduino 13. Ketika buttonState dalam kondisi LOW (off) dalam arti push buttonnya ditekan, maka led mati, selain kondisi tersebut led akan menyala terus (HIGH).

## F. LATIHAN

1. Program eksperimen dengan menggunakan 2 tombol Push Button untuk mengontrol nyala pada LED di pin 13.
  - Jika tombol 1 On, LED nyala
  - Jika tombol 2 On, LED padam
2. Membuat program eksperimen seperti gambar disamping listing program.
  - Jika saklar On, LED nyala dari 0-255 dengan step 5 kemudian dari 255-0 dengan step yang sama.
  - Jika saklar Off, LED padam.
3. Membuat program dengan ketentuan sebagai berikut:
  - Ketika saklar 1 On, LED menyala dari LED-1 menuju ke LED-4.
  - Jika saklar 2 On, maka reset (semua LED padam).
  - Jika saklar 1 On lagi, maka nyala LED menuju ke LED-4 dilaksanakan kembali.
4. Buat program untuk control kecepatan nyala LED (rotasi kanan dan kiri) menggunakan switch input, jika Sw2 On, maka delay-nya paling besar/lambat (1000ms) dan bila Sw1 On delay-nya paling kecil/cepat (100ms).



## PERCOBAAN 3

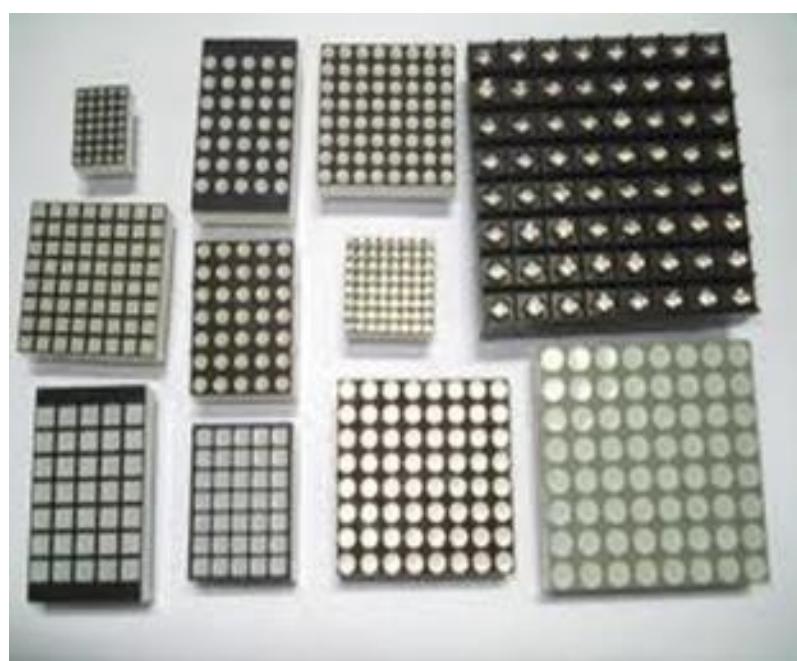
### LED DOT MATRIX

#### A. TUJUAN

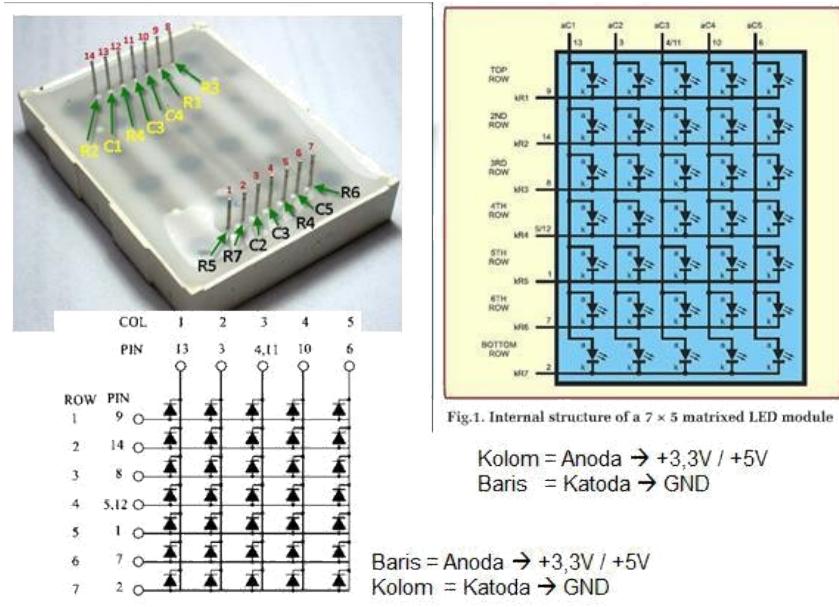
1. Membuat aplikasi system tampilan karakter yang berupa LED Dot Matrik dengan menggunakan mikrokontroler Arduino.
2. Membuat program untuk menampilkan karekter pada media LED Dot Matrik.

#### B. DASAR TEORI

Display dot matrik adalah beberapa LED yang disusun membentuk matrik baris dan kolom yang bervariasi sesuai dengan tipenya. Tipe display dot matrix bermacam-macam, salah satunya ialah tipe 5x7 atau 8x8, berupa modul jadi atau dibuat sendiri dengan menyusun beberapa LED. Ukurannya ada yang besar, sedang maupun yang kecil.



Gambar 1. Bentuk Fisik LED Dot Matrix

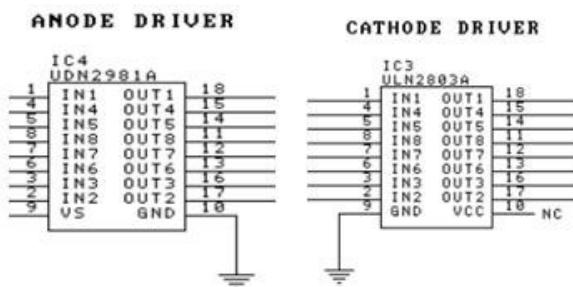


Gambar 2. Konfigurasi dan Skema LED Dot Matrix

Untuk menampilkan (*display*) dari sebuah program yang dikirim secara paralel. Misalnya,  $8 \times 8$  matriks LED Anoda bersama-sama dalam baris (R1 - R8), dan katoda dalam kolom (C1 - C8), jumlah yang diperlukan **pin I/O=16**.

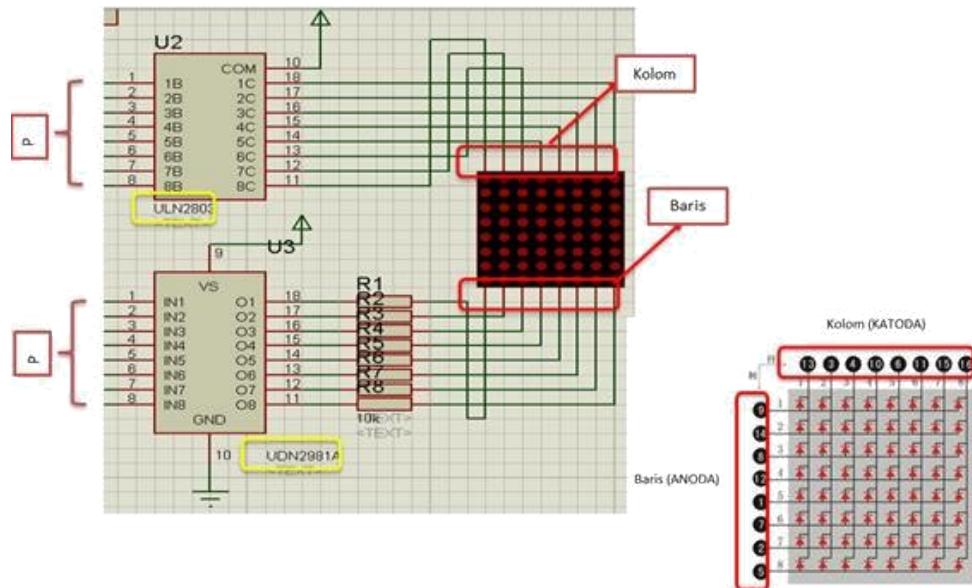
#### IC driver : UDN2981A dan ULN2803A

1. UDN2981A : driver ke transistor sekaligus mengatur anoda LED.
2. ULN2803A : driver ke transistor sekaligus mengatur katoda LED.

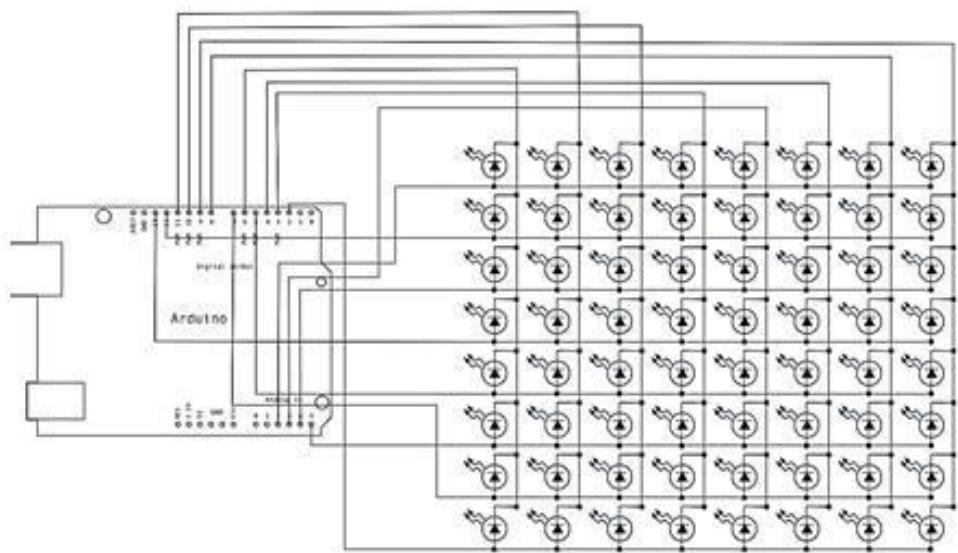


Gambar 3. Anoda Driver UDN2981A dan Cathode Driver ULN2803A

Contoh rangkaian interface menggunakan driver UDN2981A dan ULN2803A.



Gambar 4. Rangkaian Interface Menggunakan IC Driver

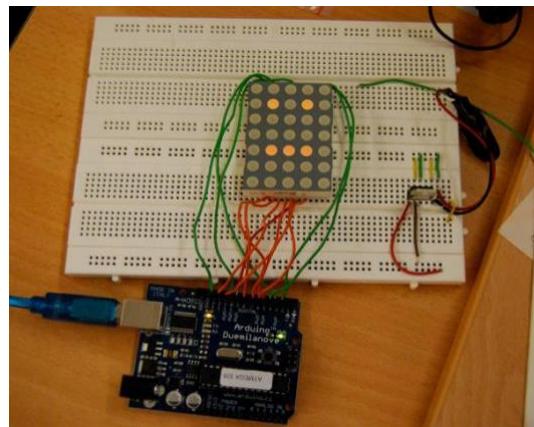


Gambar 4. Rangkaian Interface Tanpa IC Driver

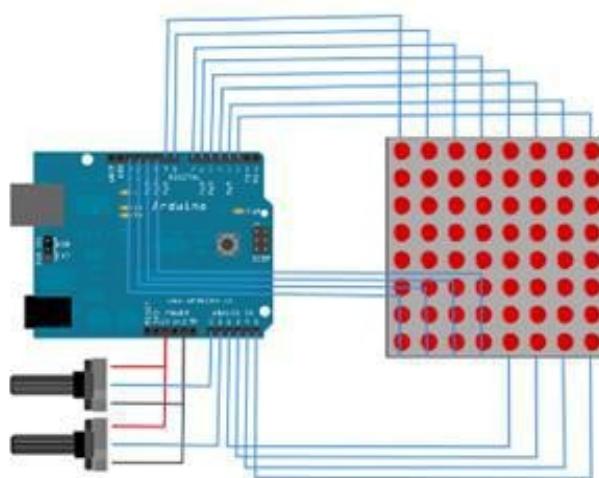
## C. HARDWARE

- Arduino Uno Board
  - LED Dot Matrix
  - Breadboard
  - Potensiometer 10Kohm
  - Kabel jumper

## D. RANGKAIAN



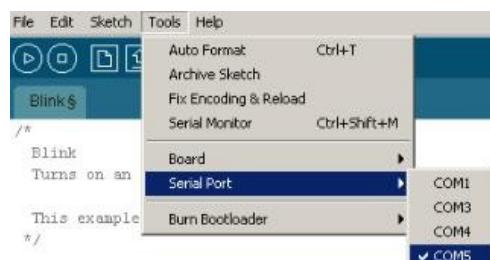
Gambar 5. Rangkaian Interface LED Dot Matrix 5x7



Gambar 6. Rangkaian Interface LED Dot Matrix 8x8

## E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke komputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



4. Jalankan arduino dan buat programnya.

## F. PROGRAM PERCOBAAN

### 1. PERCOBAAN 1

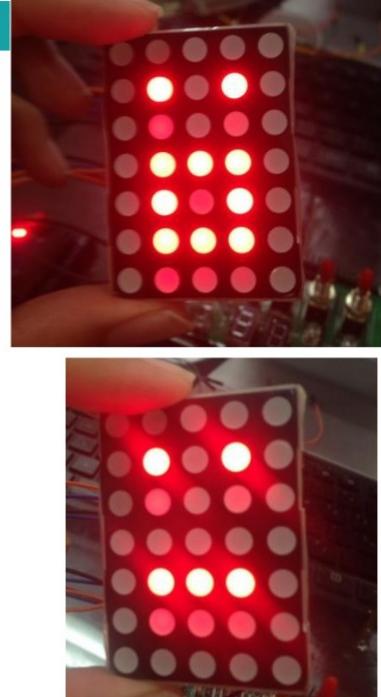
```
DOT_MATRIX

int idx = 0; unsigned long last;
void setup() {
    last = millis();
    //Kolom
    pinMode( 9, OUTPUT );
    pinMode( 10, OUTPUT );
    pinMode( 11, OUTPUT );
    pinMode( 12, OUTPUT );
    pinMode( 13, OUTPUT );
    //Baris
    pinMode( 2, OUTPUT );
    pinMode( 3, OUTPUT );
    pinMode( 4, OUTPUT );
    pinMode( 5, OUTPUT );
    pinMode( 6, OUTPUT );
    pinMode( 7, OUTPUT );
    pinMode( 8, OUTPUT );
    for( int r = 0; r < 7; r++ ) {
        digitalWrite( r + 2, LOW );
    }
    for( int c = 0; c < 5; c++ ) {
        digitalWrite( c + 9, HIGH );
    }
}
byte leds[7][5];

void setPattern( byte pattern[20][5], int idx ) {
    for( int r =0; r < 7; r++ ) {
        for( int c = 0; c < 5; c++ ) {
            leds[r][c] = pattern[r + idx][c];
        }
    }
}
void draw() {

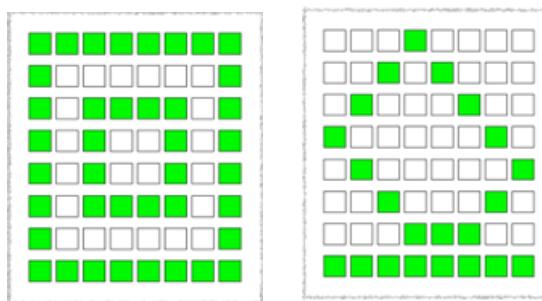
    for( int r =0; r < 7; r ++ ) {
        digitalWrite( r + 2, HIGH );
        for( int c=0; c < 5; c ++ ) {
            digitalWrite( 13 - c, ( leds[r][c] == 1 ? LOW : HIGH ) );
        }
        delayMicroseconds(900);
        digitalWrite( r + 2, LOW );
    }
}

void loop() {
    if ( millis() - last > 400 ) {
        idx = (idx == 0 ? 7 : 0);
        last = millis();
    }
}
```



## G. LATIHAN

1. Program untuk tulisan (huruf A) berjalan dari kiri ke kanan.
2. Program untuk menampilkan 2,T,A,B
3. Program satu angka paling belakang dari NRP anda menggunakan metode scanning
4. Buatlah program untuk menampilkan karakter pada display Dot Matrik menggunakan IC Driver (dikerjakan di simulator).
5. Buatlah tampilan data pada display Dot Matrik sebagai berikut;



## PERCOBAAN 4

### KONTROL NYALA LED DENGAN KEYPAD

#### A. TUJUAN

1. Melakukan proses scanning pada keypad sebagai input
2. Dapat membuat proses control display LED sebagai output

#### B. DASAR TEORI

Keypad merupakan rangkaian tombol yang berfungsi untuk member sinyal pada suatu rangkaian dengan menghubungkan jalur-jalur tertentu. Keypad terdiri dari sejumlah saklar/tombol, yang terhubung sebagai baris dan kolom. Keypad ada berbagai macam berdasarkan jumlah tombolnya. Misalnya  $3 \times 4$  dan  $4 \times 4$ . Keypad sebagai suatu input pada beberapa peralatan yang berbasis mikroprosessor atau mikrokontroller.



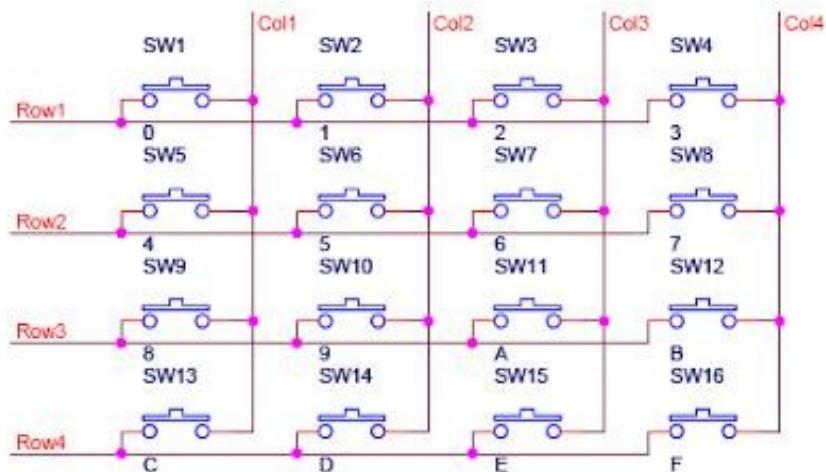
Gambar 1. Model Keypad 3x4 (Kiri) dan 4x4 (Kanan)

Agar mikrokontroller 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, handphone, dll

#### Scanning Keypad Matrix

Konfigurasi wiring diagram antara keypad matrix dan microcontroller menentukan bagaimana scanning dilakukan. Apakah dengan scanning kolom

atau baris. Setelah menentukan kolom atau baris, langkah selanjutnya kurang lebih sama. Misalkan dengan menggunakan kolom sebagai scanning. Dengan gambar contoh keypad matrix 4x4 tersebut.



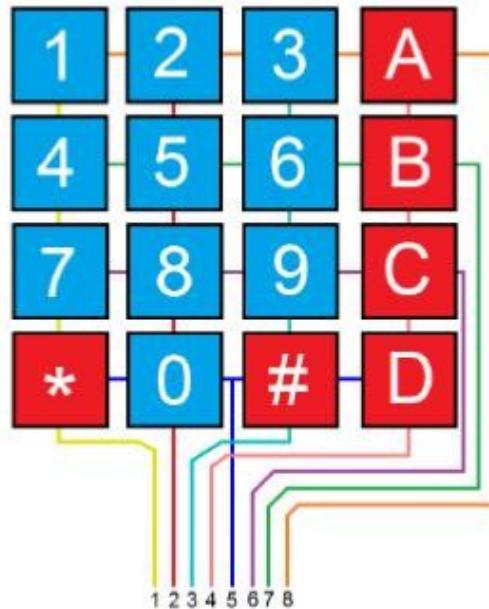
Gambar 2. Skema Rangkaian Keypad 4x4

1. Membuat definisi dari setiap tombol keypad
2. Nilai logika kolom 1, 2, 3, dan 4 akan berubah-ubah
3. Proses scanning kolom dengan hanya satu kolom yang memiliki nilai logika high
4. Pada saat kolom 1 memiliki logika high, maka kolom lainnya akan memiliki nilai logika low
5. Kondisi tersebut bergantian terus-menerus untuk semua kolom
6. Berikutnya ialah menunggu baris mana yang ditekan
7. Misal untuk menekan angka 1, baris pertama yang ditekan, dengan scanning display pada kolom 1
8. Dengan demikian, microcontroller akan mengenali input data yang ditekan kolom 1, baris 1, ialah tombol 1 yang telah didefinisikan sebelumnya

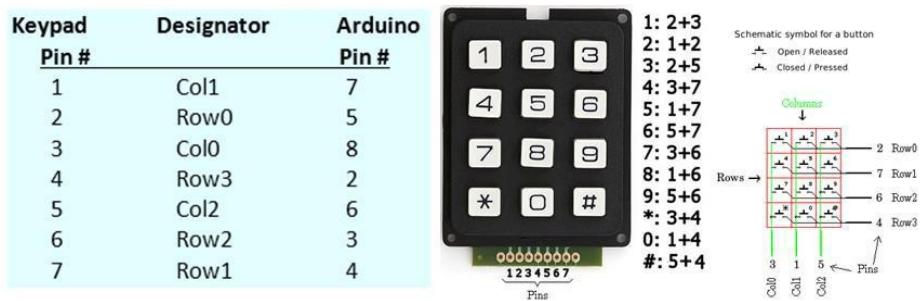
### C. HARDWARE

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

## D. RANGKAIAN



Gambar 3. Skema Koneksi Pin Keypad 4x4



Gambar 4. Tabel dan Skema Koneksi Pin Keypad 4x3

## E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke komputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



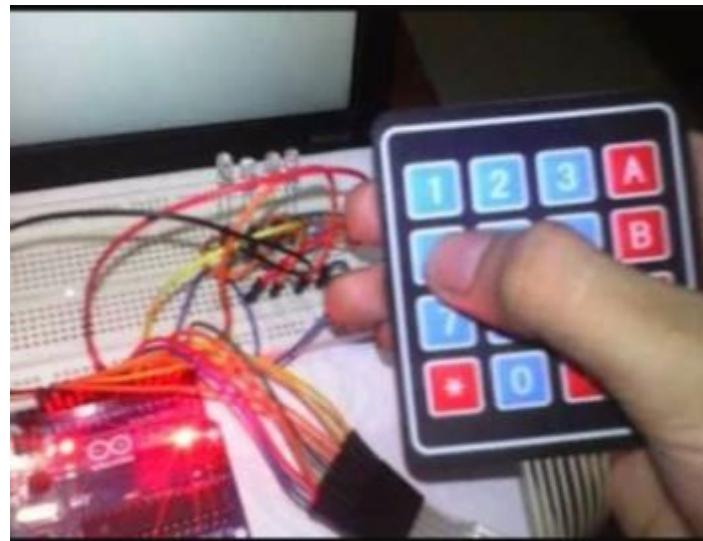
4. Jalankan arduino dan buat programnya

## F. PROGRAM

### Program-1

```
#include <Keypad.h>
const byte ROWS=4;
const byte COLS=4;
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);
byte ledPin=13;
boolean blink=false;
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);
    }
}
void keypadEvent(KeypadEvent key){
    switch(keypad.getState()){
        case PRESSED:
            switch(key){
                case '1':
                    digitalWrite(ledPin,!digitalRead(ledPin));
                    break;
                case '*':
                    digitalWrite(ledPin,!digitalRead(ledPin));
                    break;
                case RELEASED:
                    switch(key){
                        case '*':
                            digitalWrite(ledPin,!digitalRead(ledPin));
                            blink=false;
                            break;
                    }
                    break;
                case HOLD:
                    switch(key){
                        case '*':
                            blink=true;
                            break;
                    }
                    break;
            }
    }
}
```

Output:



Ketika tombol “1” atau “\*” keypad ditekan maka LED pada pin13 akan menyala, namun saat dilepas LED akan padam. Lalu ketika tombol “\*” di hold (ditekan & tahan) maka LED akan berkedip-kedip

## Program-2

```
#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);

void setup() {
    Serial.begin(9600);
}
void loop() {
    char keypressed=myKeypad.getKey();
    if(keypressed!=NO_KEY)
    {
        Serial.print(keypressed);
    }
}
```

Output:

Keypad (saat ditekan)	Serial Monitor
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
0	0
#	#
*	*

## G. LATIHAN

1. Modifikasi dari program-1 untuk menyalakan LED pada pin 13 tidak hanya menggunakan 2 tombol keypad.
2. Program untuk menyalakan LED menggunakan keypad. Misal: tombol keypad-1 (KP1) untuk menyalakan LED 1, tombol KP2 untuk LED 2 dst.

## PERCOBAAN 5

### KONTROL DISPLAY 7- SEGMENT

#### A. TUJUAN

1. Mengetahui dan memahami cara mengantarmukakan mikrokontroler dengan rangkaian penampil seven segment.
2. Mengetahui dan memahami bagaimana memprogram mikrokontroler untuk menampilkan karakter ke penampil seven segment.

#### B. DASAR TEORI

Seven Segment Display (7 Segment Display) dalam bahasa Indonesia disebut dengan Layar Tujuh Segmen adalah komponen Elektronika yang dapat menampilkan angka desimal melalui kombinasi-kombinasi segmennya. Seven Segment Display pada umumnya dipakai pada Jam Digital, Kalkulator, Penghitung atau Counter Digital, Multimeter Digital dan juga Panel Display Digital seperti pada Microwave Oven ataupun Pengatur Suhu Digital . Seven Segment Display pertama diperkenalkan dan dipatenkan pada tahun 1908 oleh Frank. W. Wood dan mulai dikenal luas pada tahun 1970-an setelah aplikasinya pada LED (Light Emitting Diode).

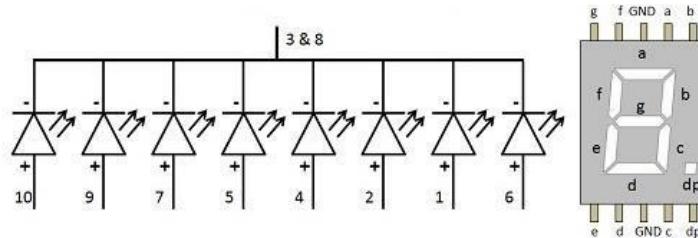
Seven Segment Display memiliki 7 Segmen dimana setiap segmen dikendalikan secara ON dan OFF untuk menampilkan angka yang diinginkan. Angka-angka dari 0 (nol) sampai 9 (Sembilan) dapat ditampilkan dengan menggunakan beberapa kombinasi Segmen. Selain 0 – 9, Seven Segment Display juga dapat menampilkan Huruf Hexadecimal dari A sampai F. Segmen atau elemen-elemen pada Seven Segment Display diatur menjadi bentuk angka “8” yang agak miring ke kanan dengan tujuan untuk mempermudah pembacaannya. Pada beberapa jenis Seven Segment Display, terdapat juga penambahan “titik” yang menunjukan angka koma decimal. Terdapat beberapa jenis Seven Segment Display, diantaranya adalah Incandescent bulbs, Fluorescent lamps (FL), Liquid Crystal Display (LCD) dan Light Emitting Diode (LED).

Salah satu jenis Seven Segment Display yang sering digunakan oleh para penghobi Elektronika adalah 7 Segmen yang menggunakan LED (Light Emitting Diode) sebagai penerangnya. LED 7 Segmen ini umumnya memiliki 7 Segmen atau elemen garis dan 1 segmen titik yang menandakan “koma” Desimal. Jadi Jumlah keseluruhan segmen atau elemen LED sebenarnya adalah 8. Cara kerjanya pun boleh dikatakan mudah, ketika segmen atau elemen tertentu diberikan arus listrik, maka Display akan menampilkan angka atau digit yang diinginkan sesuai dengan kombinasi yang diberikan.

Terdapat 2 Jenis LED 7 Segmen, diantaranya adalah “LED 7 Segmen common Cathode” dan “LED 7 Segmen common Anode”.

### 1. LED 7 Segmen Tipe Common Cathode (Katoda)

Pada LED 7 Segmen jenis Common Cathode (Katoda), Kaki Katoda pada semua segmen LED adalah terhubung menjadi 1 Pin, sedangkan Kaki Anoda akan menjadi Input untuk masing-masing Segmen LED. Kaki Katoda yang terhubung menjadi 1 Pin ini merupakan Terminal Negatif (-) atau Ground sedangkan Signal Kendali (Control Signal) akan diberikan kepada masing-masing Kaki Anoda Segmen LED. LED Seven Segment Display Tipe Common Katoda.

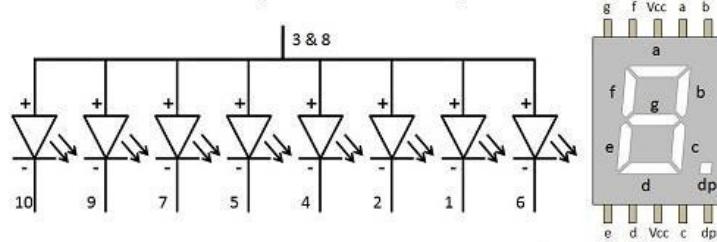


Gambar 1. 7-Segment Tipe Common Cathode

### 2. LED 7 Segmen Tipe Common Anode (Anoda)

Pada LED 7 Segmen jenis Common Anode (Anoda), Kaki Anoda pada semua segmen LED adalah terhubung menjadi 1 Pin, sedangkan kaki Katoda akan menjadi Input untuk masing-masing Segmen LED. Kaki Anoda yang terhubung menjadi 1 Pin ini akan diberikan Tegangan Positif

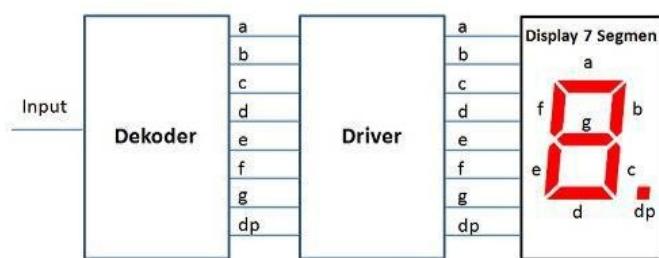
( ) dan Signal Kendali (control signal) akan diberikan kepada masing-masing Kaki Katoda Segmen LED. LED Seven Segment Display Tipe Common Anoda.



Gambar 2. 7-Segment Tipe Common Anode

### Prinsip Kerja Dasar Driver System pada LED 7 Segmen

Blok Dekoder pada diagram diatas mengubah sinyal Input yang diberikan menjadi 8 jalur yaitu “a” sampai “g” dan poin decimal (koma) untuk meng-ON-kan segmen sehingga menghasilkan angka atau digit yang diinginkan. Contohnya, jika output dekoder adalah a, b, dan c, maka Segmen LED akan menyala menjadi angka “7”. Jika Sinyal Input adalah berbentuk Analog, maka diperlukan ADC (Analog to Digital Converter) untuk mengubah sinyal analog menjadi Digital sebelum masuk ke Input Dekoder. Jika Sinyal Input sudah merupakan Sinyal Digital, maka Dekoder akan menanganinya sendiri tanpa harus menggunakan ADC.



Gambar 3. Blok Diagram Dasar 7-Segment

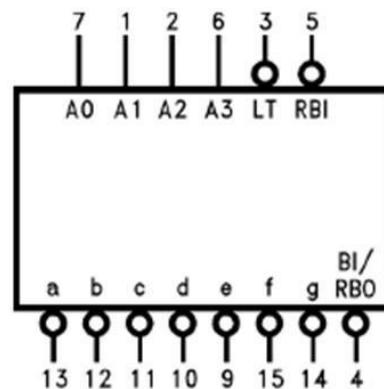
Fungsi daripada Blok Driver adalah untuk memberikan arus listrik yang cukup kepada Segmen/Elemen LED untuk menyala. Pada Tipe Dekoder tertentu, Dekoder sendiri dapat mengeluarkan Tegangan dan Arus listrik yang cukup untuk menyalakan Segmen LED maka Blok Driver ini tidak

diperlukan. Pada umumnya Driver untuk menyalakan 7 Segmen ini adalah terdiri dari 8 Transistor Switch pada masing-masing elemen LED. Dekoder BCD ke 7-Segment ada dua yang sering atau umum digunakan, yaitu :

- IC TTL 74LS47 menyalakan 7 segment mode common anoda.
- IC TTL 74LS48 menyalakan 7 segment mode common katoda.

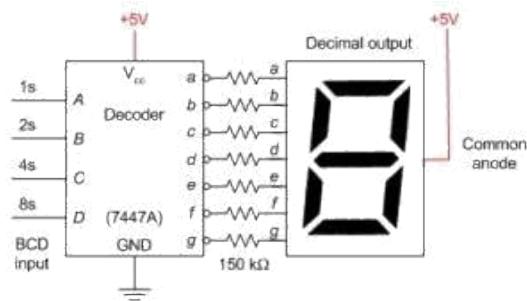
### Konfigurasi Pin IC TTL 74LS47 dan 74LS48

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, sedangkan untuk dekoder 7448 adalah aktif high. Tiga masukan ekstra juga ditunjukkan pada konfigurasi pin IC 7447 atau IC 7448 yaitu lamp test, blanking input/ripple blanking output, dan ripple blanking input.

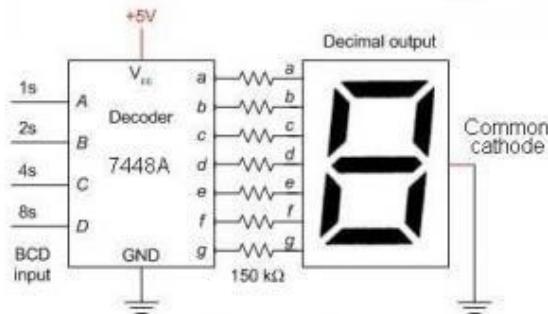


Gambar 4. Konfigurasi Pin IC TTL 74LS47

- **Lamp Test (LT)**, 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.



Gambar 5. Rangkaian Aplikasi Dekoder BCD Ke 7 Segmen Common Anoda (IC 7447)

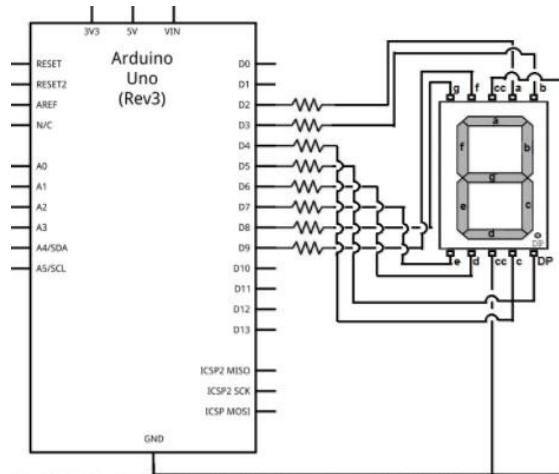


Gambar 6. Rangkaian Aplikasi Dekoder BCD Ke 7 Segmen Common Anoda (IC 7448)

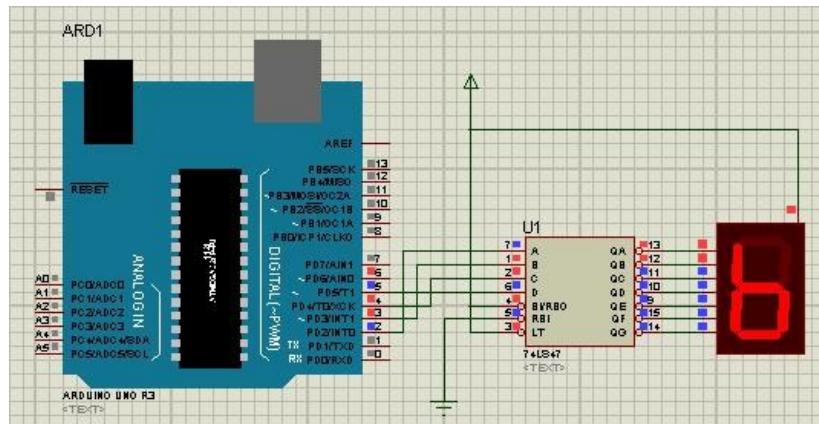
### C. HARDWARE

- Arduino Uno Board
- 7-Segment
- Resistor 330 ohm (bila diperlukan)
- Breadboard
- Kabel jumper

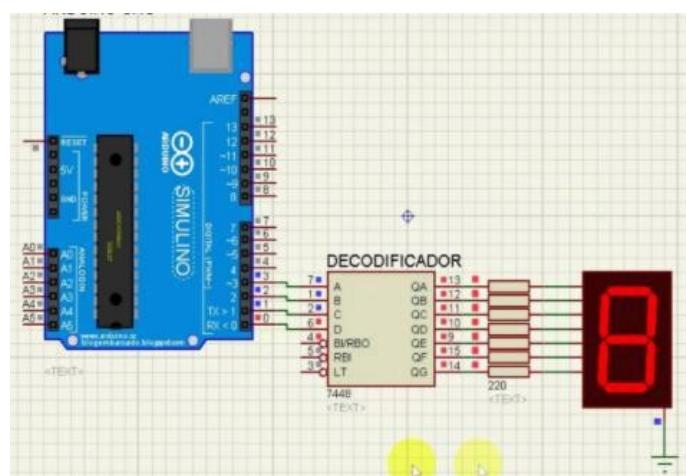
## D. RANGKAIAN



Gambar 7. Skema Koneksi Arduino ke 7-Segment



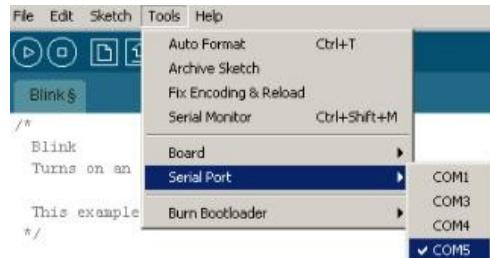
Gambar 8. Skema Koneksi Arduino ke 7-Segment Dengan Dekoder IC 74LS47



Gambar 8. Skema Koneksi Arduino ke 7-Segment Dengan Dekoder IC 74LS48

## E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke komputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



4. Jalankan arduino dan buat programnya

## F. PROGRAM

1. Nyala 7 segment dari 0 sampai 9

```
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);
};
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);
};
```

```

    - 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);
    };
    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;
        };
    };
    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); };
        };
    };
}

```

2. Nyala 7 segmen dari 0 ke 9 dan kembali dari 9 ke 0 (common anoda)

```
//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
{ 0,0,0,1,1,0,0 }, // = 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
};

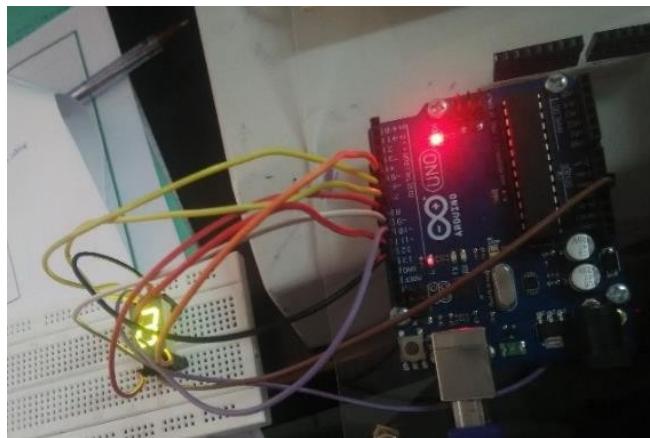
void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  writeDot(1); // padamkan tanda "dot" (titik)
}
void writeDot(byte dot) {
  digitalWrite(6, dot);
}
void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

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);
  }
}
```

HASIL :



## PERCOBAAN 6

### PEMROGRAMAN LCD

#### A. TUJUAN

1. Mengetahui dan memahami cara memprogram mikrokontroler dengan modul penampil LCD.
2. Membuat program untuk menampilkan karakter pada LCD.

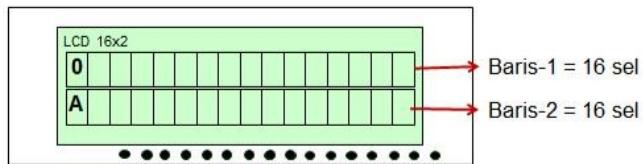
#### B. DASAR TEORI

LCD (Liquid Crystal Display) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang, misalnya dalam alat-alat elektronik, seperti televisi, kalkulator ataupun layar computer.



Gambar 1. Macam – Macam LCD

Pada Percobaan kali ini adalah dengan menggunakan LCD 16x2 yang artinya LCD tersebut terdiri dari 16 kolom dan 2 baris karakter (tulisan). Modul LCD yang dipakai ialah Hitachi HD44780U Dot Matrix LCD, dapat menampilkan data karakter alphanumerik yang dibangkitkan dari generator RAM. Generator RAM sebagai driver dot-matrik LCD. Karakter dalam bentuk ukuran 5x8 / 5x11 dot matrik. Dalam modul LCD terdapat microcontroller yang berfungsi sebagai pengendali tampilan karakter LCD. Microntroller pada LCD dilengkapi dengan memori dan register.



Gambar 2. Mikroprosessor dapat mengendalikan LCD melalui instruction register (IR) dan data register (DR).

Kemudian ada 3 jenis memori microcontroler dalam internal LCD, yaitu :

1. *DDRAM (Display Data Random Access Memory)* : tempat karakter yang akan ditampilkan,
2. *CGRAM (Character Generator Random Access Memory)* : memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.
3. *CGROM (Character Generator Read Only Memory)* : memori untuk menggambarkan pola sebuah karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD.

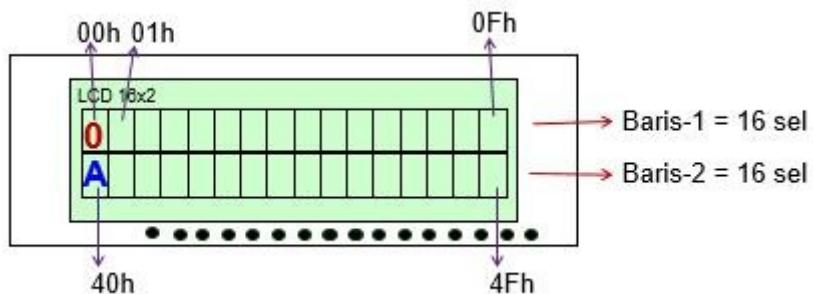
Konfigurasi alamat DDRAM untuk dua baris :

- Display baris I : 00h – 27h
- Display baris II : 40h – 67h

LCD 16 x 2, baris I alamat DDRAM 00h – 0Fh dan baris II adalah alamat 40h – 4Fh. Jika display digeser ke kanan satu kali, maka yang tampil pada baris I adalah alamat 01h – 10h dan baris II adalah alamat 41h – 50h.

**Contoh:**

Karakter „A“ atau 41H ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila ditulis di alamat 40, maka karakter tersebut akan tampil pada baris kedua, kolom pertama dari LCD. Jika display digeser ke kanan satu kali, maka yang tampil pada baris I adalah alamat 01h – 10h dan baris II adalah alamat 41h – 50h.



Gambar 3. Ilustrasi Konfigurasi Alamat DDRAM Pada LCD

Tidak hanya memori saja, LCD juga mempunyai register control, diantaranya adalah :

1. Register perintah : berisi perintah-perintah dari mikrokontroler ke panel LCD pada saat proses penulisan data atau tempat status dari panel LCD dapat dibaca pada saat pembacaan data.
2. Register data : untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

No.	Instruction	Hex	Decimal
<b>1</b>	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
<b>2</b>	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
<b>3</b>	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
<b>4</b>	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
<b>5</b>	Entry Mode	0x06	6
<b>6</b>	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
<b>7</b>	Display on Cursor on	0x0E	14
<b>8</b>	Display on Cursor off	0x0C	12
<b>9</b>	Display on Cursor blinking	0x0F	15
<b>10</b>	Shift entire display left	0x18	24
<b>12</b>	Shift entire display right	0x1C	30
<b>13</b>	Move cursor left by one character	0x10	16
<b>14</b>	Move cursor right by one character	0x14	20
<b>15</b>	Clear Display (also clear DDRAM content)	0x01	1
<b>16</b>	Set DDRAM address or cursor position on display	0x80+add	128+add

Tabel 1. Instruksi Control LCD

### Jalur input dan kontrol LCD

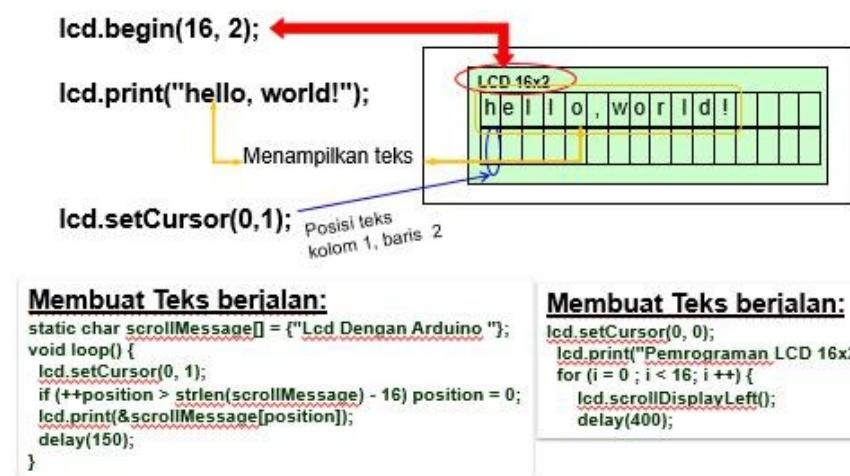
1. *Pin data* : jalur untuk memberikan data karakter yang ingin ditampilkan pada LCD, dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
2. *Pin RS (Register Select)* : berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukkan yang masuk adalah *perintah*, sedangkan logika *high* menunjukkan *data*.
3. *Pin R/W (Read Write)* : berfungsi sebagai instruksi pada modul jika *low tulis data*, sedangkan *high baca data*.

4. *Pin E (Enable)* : mengaktifkan LCD pada proses penulisan data ke Register Kontrol dan Register Data LCD.
5. *Pin VLCD* : berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.



Gambar 4. Jalur input dan kontrol LCD

#### Penulisan program :

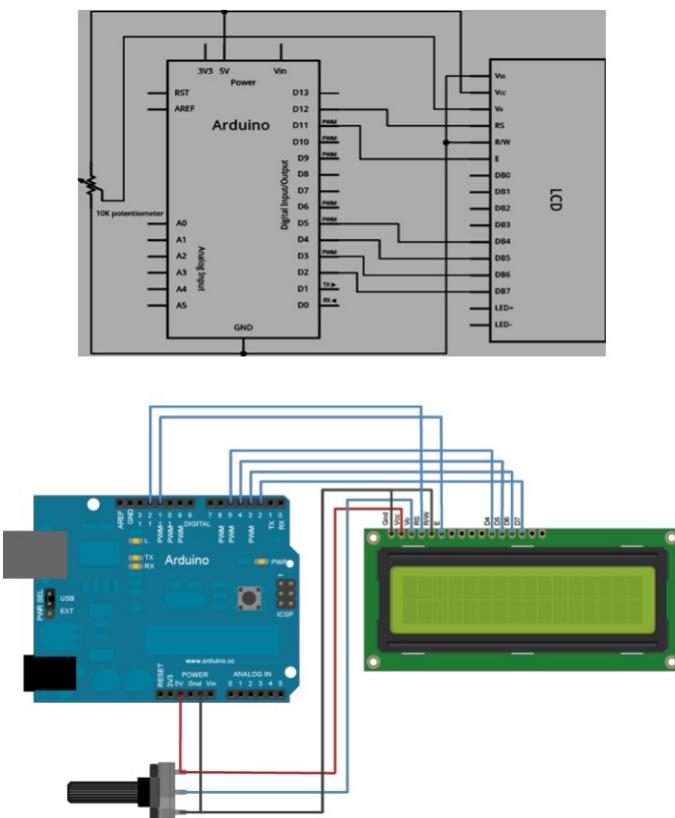


Gambar 5. Penulisan Program Pada LCD Menggunakan Arduino

### C. HARDWARE

- Arduino Uno Board
- 1 buah LCD 16x2 Hitachi HD44780
- 1 buah potensiometer / variabel resistor 10K
- Kabel jumper

### D. RANGKAIAN



Gambar 3. Skema Koneksi Arduino ke LCD 16x2

### E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke komputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



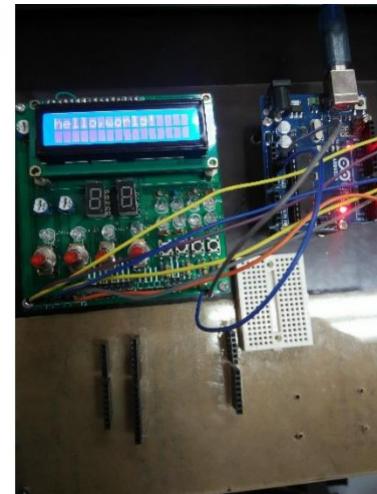
4. Jalankan arduino dan buat programnya

## F. PROGRAM

1. Menampilkan tulisan “hello, world!” berkedip dengan durasi waktu 500ms

```
program1
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
|
void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}
void loop() {
  lcd.noDisplay();
  delay(500);
  lcd.display();
  delay(500);
}
```

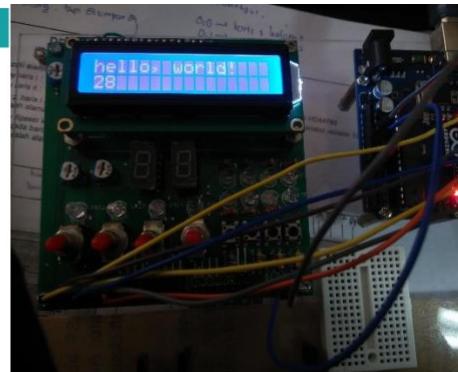


2. Baris pertama menampilkan tulisan “hello,world” dan baris kedua mengitung angka dari 0-tak hingga secara up counter dengan durasi 1000ms setiap ganti angka

```
program2
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}
void loop() {
  lcd.setCursor(0,1);
  lcd.print(millis() / 1000);
}
```



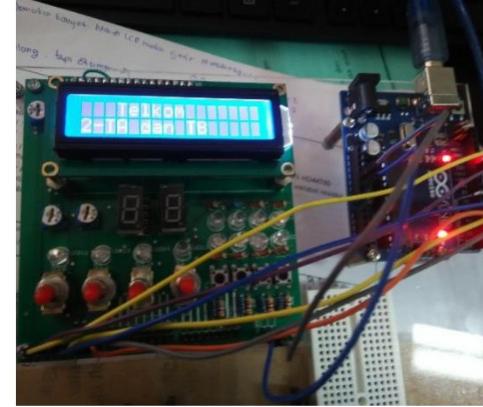
3. Menampilkan tulisan “Telkom” pada baris pertama dan menampilkan “2-TA dan TB” pada baris kedua

```
program3
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
    lcd.begin(16, 2);
    lcd.setCursor(3,0);
    lcd.print("Telkom");
    lcd.setCursor(0,1);
    lcd.print("2-TA dan TB");
}

void loop()
{}
```

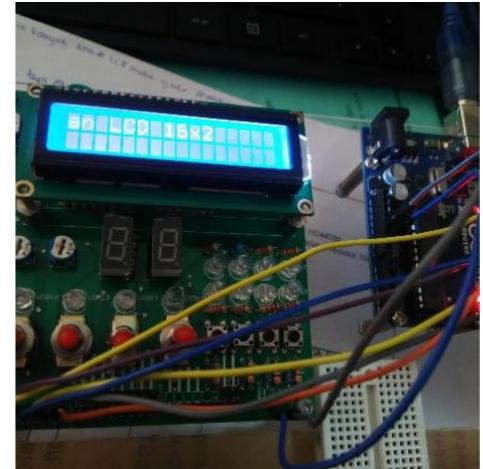


4. Menampilkan tulisan “Pemrograman LCD 16x2” dan bergeser dari kanan ke kiri

```
program4
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    lcd.begin(16, 2);
}
void loop() {
    int i;
    lcd.setCursor(0, 0);
    lcd.print("Pemrograman LCD 16x2");
    for (i = 0 ; i < 16; i++) {
        lcd.scrollDisplayLeft();
        delay(400);
    }
}
```



5. Menampilkan tulisan “Belajar Program” pada baris pertama di tengah dan baris kedua menampilkan “LCD Dengan Arduino” bergeser dari kanan kekiri

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
static char scrollMessage[] = "      LCD Dengan Arduino      ";
int position = 0;
void setup() {
    lcd.begin(16, 2);
    lcd.print("Belajar Program");
}
void loop() {
    lcd.setCursor(0, 1);
    if (++position > strlen(scrollMessage) - 16) position = 0;
    lcd.print(scrollMessage[position]);
    delay(150);
}
```



## G. LATIHAN

1. Modifikasi program contoh diatas supaya dapat menampilkan data hello, world! di baris-1, 5171 di baris-2 pada LCD dan bergerak kekanan kekiri 3x.



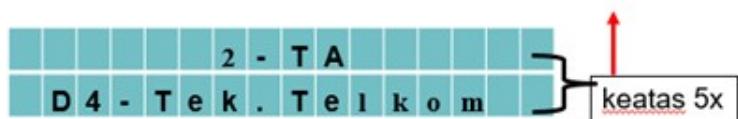
2. Buat program untuk menampilkan tulisan kelas anda diberis pertama LCD dan tulisan prodi anda di baris kedua LCD
  - tulisan di baris 1 dan 2 berjalan/ bergeser bersama-sama kekanan 3x dan kiri 3x kemudian berkedip 3x
3. Buat program untuk menampilkan tulisan kelas anda ditengah baris pertama LCD kemudian tulisan berjalan berputar seperti pada gambar berikut.



4. Buat program untuk menampilkan tulisan HALO di baris pertama LCD kemudian tulisan berjalan zig-zag seperti pada gambar dibawah ini.



5. Buat program untuk menampilkan tulisan kelas anda diberis pertama LCD dan tulisan prodi anda di baris kedua LCD . Tulisan baris-1 dan 2 bergerak keatas 5x



## PERCOBAAN 7

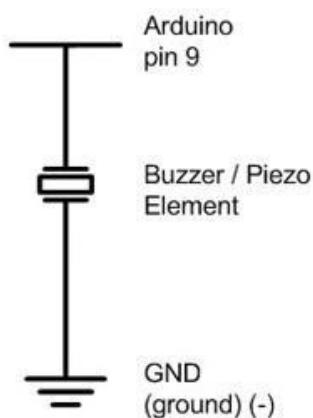
### PEMROGRAMAN MELODY

#### A. TUJUAN

1. Membuat melody Do, Re, Mi, Fa, Sol, La, Si, Do.
2. Membuat program untuk membangkitkan frekuensi nada melody

#### B. DASAR TEORI

Suara adalah fenomena analog, bagaimana caranya Arduino menanganinya? Sekali lagi kita akan meniru perilaku analog menggunakan Arduino dengan kecepatannya yang luar biasa. Untuk ini kita akan menghubungkan buzzer/piezo elements ke salah satu pin digital Arduino. Sebuah buzzer akan menghasilkan suara ketukan setiap kali dialiri gelombang (pulse) arus listrik. Jika kita menggunakan gelombang arus listrik dengan frekuensi yang tepat (misalnya 440 kali per detik untuk menghasilkan nada A) maka suara ketukan secara bersama-sama ini akan menghasilkan nada musik.



Gambar 1. Skema Rangkaian Koneksi Arduino Dengan Buzzer

Berikut merupakan tabel frekuensi yang dapat digunakan untuk menghasilkan nada sesuai dengan yang diinginkan.

NADA	DO	RE	MI	FA	SOL	LA	SI	DO"
<b>FREKUENSI</b>	262	294	330	349	395	440	494	523

note	frequency	period	timeHigh
c	261 Hz	3830	1915
d	294 Hz	3400	1700
e	329 Hz	3038	1519
f	349 Hz	2864	1432
g	392 Hz	2550	1275
a	440 Hz	2272	1136
b	493 Hz	2028	1014
C	523 Hz	1912	956

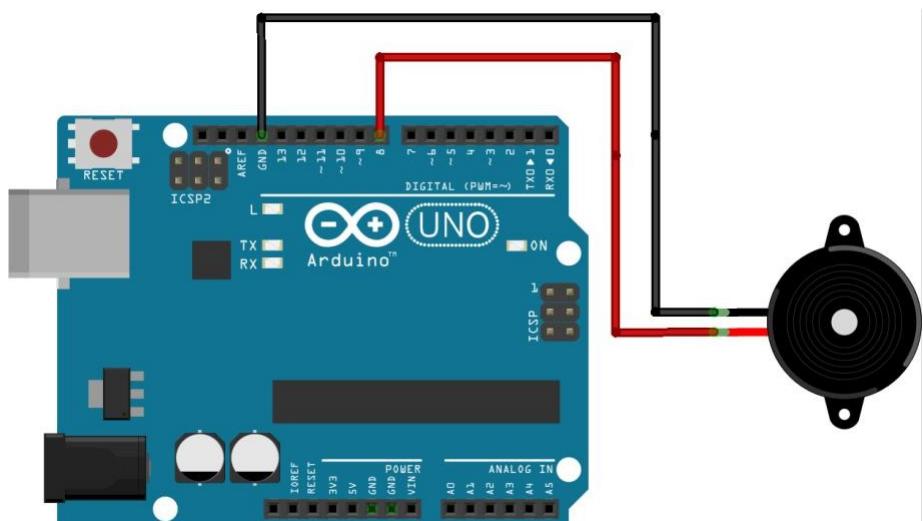
$$\text{timeHigh} = \text{period} / 2 = 1 / (2 * \text{toneFrequency})$$

Tabel 1. Tabel Frekuensi Untuk Arduino

### C. HARDWARE

- Arduino Uno Board
- 1 buah Piezo Speaker
- Kabel jumper

### D. RANGKAIAN



Gambar 2. Skema Koneksi Arduino ke Buzzer

## E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke komputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



4. Jalankan arduino dan buat programnya

## F. PROGRAM

### 1. Bunyi DO-RE-MI-FA-SOL-LA-SI-DO

```
melody
int speaker= 9;

void setup() { }

void loop() {
    tone(speaker, 262);    delay(500); // nada DO
    tone(speaker, 294);    delay(550); // nada RE
    tone(speaker, 330);    delay(590); // nada MI
    tone(speaker, 349);    delay(600); // nada FA
    tone(speaker, 395);    delay(610); // nada SOL
    tone(speaker, 440);    delay(620); // nada LA
    tone(speaker, 494);    delay(630); // nada SI
    tone(speaker, 523);    delay(700); // nada DO
}
```

### 2. Bunyi yang dihasilkan berupa lagu Twinkle Little Star

```
melody1
int speakerPin = 9;

int length = 15;
char notes[] = "ccggaagffeeddc ";
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;
void playTone(int tone, int duration) {
    for (long i = 0; i < duration * 1000L; i += tone * 2) {
        digitalWrite(speakerPin, HIGH);
        delayMicroseconds(tone);
        digitalWrite(speakerPin, LOW);
        delayMicroseconds(tone);
    }
}
```

```

void playNote(char note, int duration) {
    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
    int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014,
    956 };

    // play the tone corresponding to the note name
    for (int i = 0; i < 8; i++) {
        if (names[i] == note) {
            playTone(tones[i], duration);
        }
    }
}

void setup() {
    pinMode(speakerPin, OUTPUT);
}

void loop() {
    for (int i = 0; i < length; i++) {
        if (notes[i] == ' ') {
            delay(beats[i] * tempo); // rest
        } else {
            playNote(notes[i], beats[i] * tempo);
        }
        // pause between notes
        delay(tempo / 2);
    }
}

```

3. Bunyi yang dihasilkan berupa lagu “Lihat Kebunku”

```

melody2

int speaker= 9;
void setup(){ }
void DO() { tone(speaker, 262); }
void RE() { tone(speaker, 294); }
void MI() { tone(speaker, 330); }
void FA() { tone(speaker, 349); }
void SOL() { tone(speaker, 395); }
void LA() { tone(speaker, 440); }
void SI() { tone(speaker, 494); }
void DOO() { tone(speaker, 523); }
void DIAM() { noTone(speaker); }
void loop() {
    SOL(); delay(200); DO(); delay(200); DO(); delay(200);
    SI(); delay(200); DO(); delay(200); RE(); delay(200);
    SOL(); delay(200); SOL(); delay(200);
    RE(); delay(200); RE(); delay(200); DO(); delay(200);
    RE(); delay(200); MI(); delay(200);
    FA(); delay(200); RE(); delay(200); RE(); delay(200);
    MI(); delay(200); DO(); delay(200); DO(); delay(200);
    RE(); delay(200); RE(); delay(200); DO(); delay(200);
    SI(); delay(200); DOO(); delay(200); DIAM(); delay(1000);
    FA(); delay(200); RE(); delay(200); RE(); delay(200);
    MI(); delay(200); DO(); delay(200); DO(); delay(200);
    RE(); delay(200); RE(); delay(200); DO(); delay(200);
    SI(); delay(200); DOO(); delay(200);
}

```

## **G. LATIHAN**

1. Buatlah eksperimen dengan menggunakan kombinasi nada-nada yang lain disertai dengan tampilan LED.
2. Buatlah eksperimen dengan menggunakan kombinasi nada-nada yang membentuk sebuah lagu.
  - Membunyikan lagu “Happy Birthday”.
  - Membunyikan lagu “Es Krim Walls”.
  - dll

## PERCOBAAN 8

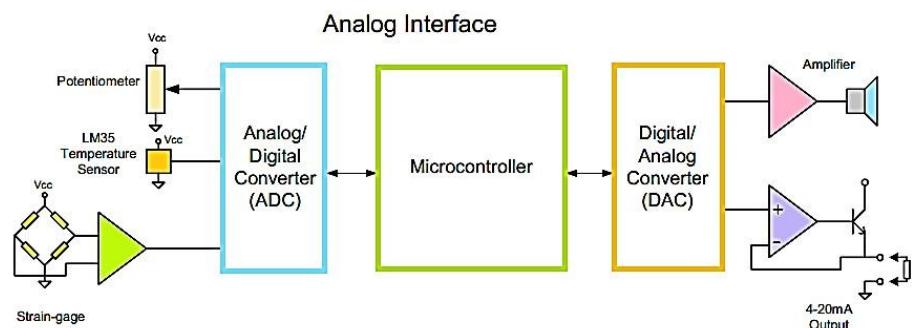
### ANALOG INPUT (ADC)

#### A. TUJUAN

1. Membaca tegangan analog yang masuk pada pin analog Arduino
2. Membuat program untuk proses konversi dari nilai analog menjadi digital atau ADC (*Analog to Digital Conversion*)

#### B. DASAR TEORI

Pada percobaan kali ini membahas mengenai bagaimana cara menghitung nilai pembacaan ADC yang berasal dari suatu sensor analog. Microcontroller memiliki kemampuan untuk menerima, mengolah, dan memberikan output signal digital. Sebagai contoh bila microcontroller dengan catu daya 5 volt, microcontroller tersebut dapat mengenali tegangan sebesar 0 volt sebagai signal digital low, atau tegangan 5 volt sebagai signal digital high. Sedangkan untuk mengenali nilai tegangan antara 0 volt hingga 5 volt, diperlukan feature khusus, yakni Analog-to-Digital Converter. Analog-to-Digital Converter atau biasa disebut ADC, memungkinkan microcontroller untuk mengenali suatu nilai analog melalui suatu pendekatan digital. Saat ini banyak microcontroller, termasuk Arduino telah dilengkapi dengan feature ADC yang terintegrasi di dalamnya.



Gambar 1. Contoh Antarmuka ADC Mikrokontroller

Tapi tidak semua pin Arduino dapat digunakan untuk mengolah signal analog. Pada board Arduino Uno, terdapat enam pin analog, yakni mulai dari

A0 hingga A5. Huruf A pada awal nama pin Arduino menandakan pin tersebut dapat digunakan untuk mengolah signal analog. Seberapa tepat nilai signal analog yang dipetakan secara digital, ditentukan oleh seberapa besar resolusi ADC. Semakin besar resolusi ADC, maka semakin mendekati nilai analog dari signal tersebut. Untuk resolusi ADC pada board Arduino Uno ialah 10 bit, yang berarti mampu memetakan hingga 1024 discrete analog level. Beberapa jenis microcontroller lain memiliki resolusi 8 bit, 256 discrete analog level, bahkan ada yang memiliki resolusi 16 bit, 65536 discrete analog level.



Gambar 2. Pin Analog Pada Arduino Yang Bisa Digunakan Untuk Memproses Data Analog

### Kaitan Nilai ADC dengan Tegangan

Nilai ADC menunjukkan ratio perbandingan dengan tegangan yang terbaca. Berikut persamaannya ialah nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber. Nilai ADC tergantung dengan tegangan yang menjadi catu daya sistem microcontroller. Untuk board Arduino biasa menggunakan sumber tegangan 5 volt. Berikut adalah cara mencari nilai ADC, dengan menerapkan persamaan yang ada dan tegangan terbaca sebesar 2,12 volt pada board Arduino Uno.

- Board Arduino Uno memiliki resolusi 10 bit, dengan nilai terbesar 1023
- Tegangan sumber 5 volt dan tegangan terbaca ialah 2,12 volt
- Nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber
- Sehingga diperoleh nilai ADC sebesar 434
- 

Dengan demikian diperoleh nilai ADC sebesar 434 dari tegangan terukur 2,12 volt. Untuk setiap unit ADC tersebut memiliki perbandingan tegangan sebesar 4,9 mV. Selain dipengaruhi oleh besarnya nilai resolusi ADC, tepat tidaknya pengukuran nilai ADC juga dipengaruhi oleh clock speed ADC tersebut. Untuk board Arduino Uno sendiri clock speed ADC maximum yang disarankan ialah 200 kHz. Nilai clock speed 200 kHz tersebut berdasarkan spesifikasi internal DAC (Digital to Analog Converter) pada rangkaian pengubahnya. Meski demikian, penggunaan clock speed pada 1 MHz mesalnya, tidak mengurangi kualitas resolusi ADC tersebut.

### Cara Mencari Nilai ADC

$$\frac{\text{Resolution of the ADC}}{\text{System Voltage}} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

$$\frac{1023}{5} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

Jika Tegangan Analog **2.12 Volt**, maka nilai ADC:

$$\frac{1023}{5V} = \frac{X}{2.12V}$$

$$\frac{1023}{5V} * 2.12V = X$$

$$X = 434$$

Jika tegangan 5 volt dikonversi menjadi data digital 10 bit

$$\frac{5}{1023} = 0,004887585$$

Artinya setiap 1 angka desimal mewakili tegangan sebesar 0,004887585 volt.

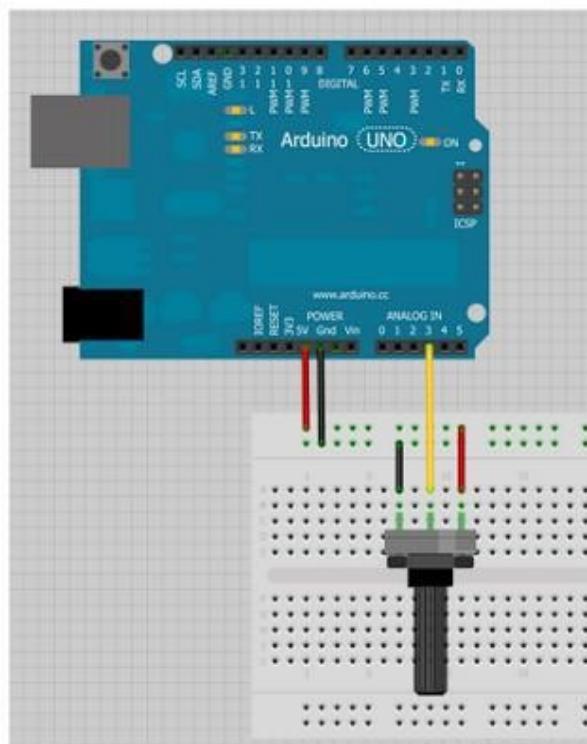
Berapa besar tegangan yang diwakili angka 512?

$$\frac{5V}{1023} * 512 = 2,50244 \text{ Volt}$$

### C. HARDWARE

- Arduino Uno Board
- 1 buah Piezo Speaker
- Kabel jumper

### D. RANGKAIAN



Gambar 3. Skema Koneksi Arduino ke pin Analog Dengan Variabel Resistor Untuk Membaca Nilai Data Analog.

### E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke komputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



4. Jalankan arduino dan buat programnya

## F. PROGRAM

1. Percobaan 1

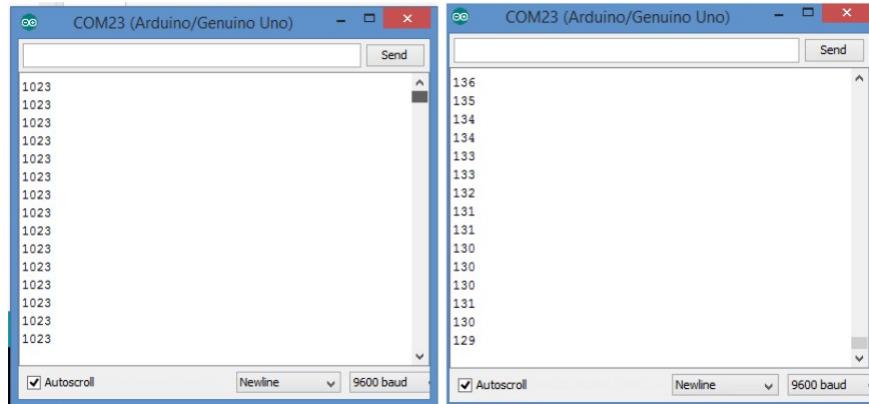
Menyalakan led dengan kontrol potensiometer, jika diputar ke kanan led akan berkedip dengan cepat dan jika diputar ke kiri led berkedip lambat

```
adc1
int sensorPin = A0; // pilih pin input untuk potentiometer
int ledPin = 13; // pilih pin untuk LED
int sensorValue = 0; // variabel simpan nilai dari sensor
void setup() {
  pinMode(ledPin, OUTPUT); // deklarasi ledPin untuk OUTPUT
}
void loop() {
  sensorValue = analogRead(sensorPin); // baca data dari sensor
  digitalWrite(ledPin, HIGH); // ledPin on
  delay(sensorValue); // stop program u/ <sensorValue> milidetik
  digitalWrite(ledPin, LOW); // ledPin off
  delay(sensorValue);
}
```

2. Percobaan 2a

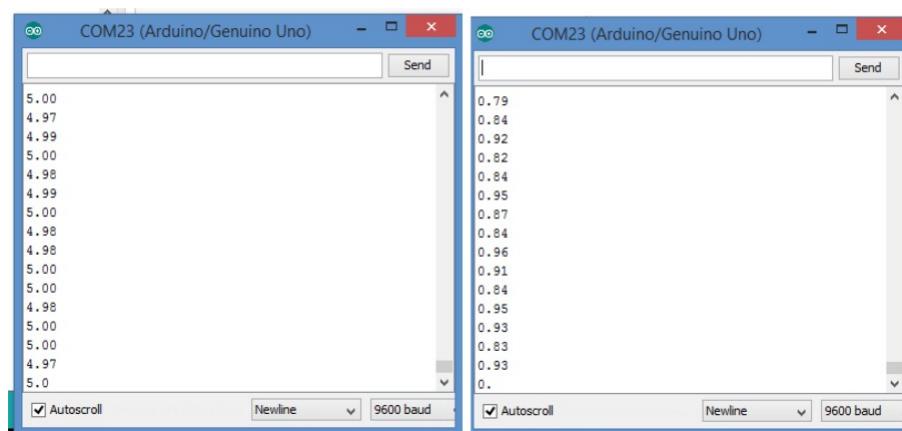
Membaca tegangan adc di serial monitor jika potensiometer diputar ke kiri angka akan semakin kecil dan jika diputar ke kanan angka akan semakin tinggi 0-1023

```
adc2a
int analogPin = A0;
int lang = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  lang = analogRead(analogPin);
  delay(100);
  Serial.println(lang);
}
```



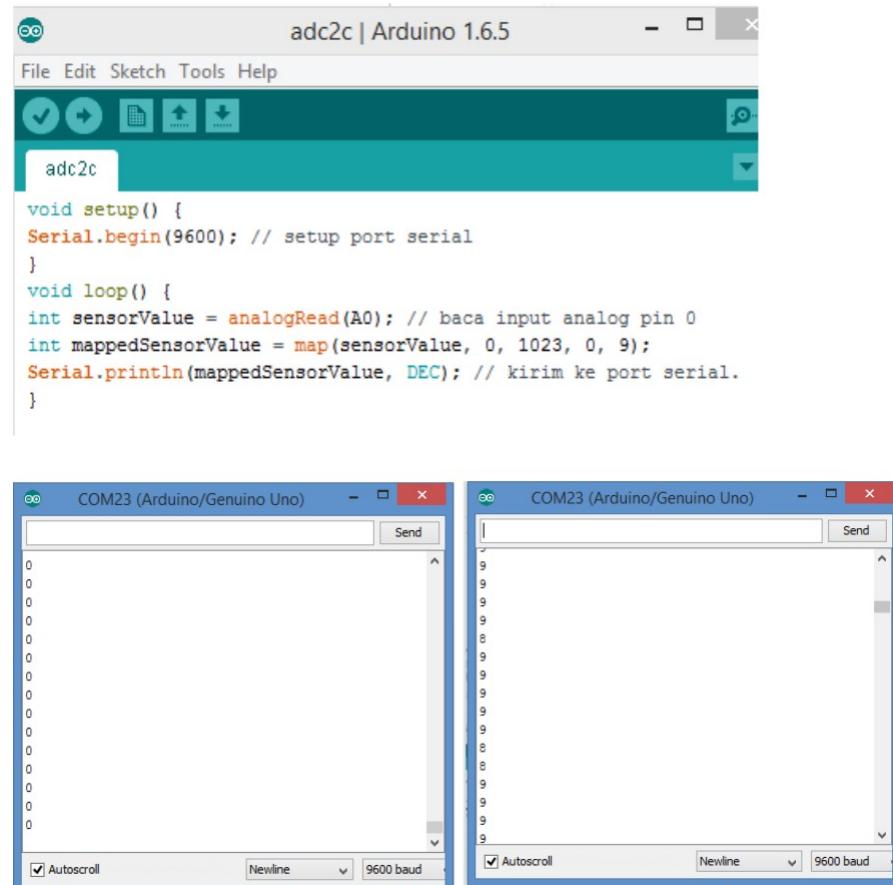
### 3. Percobaan 2b

Membaca tegangan adc di serial monitor jika potensiometer diputar ke kiri angka akan semakin kecil dan jika diputar ke kanan angka akan semakin tinggi 0-5



#### 4. Percobaan 2c

Membaca tegangan adc di serial monitor jika potensiometer diputar ke kiri angka akan semakin kecil dan jika diputar ke kanan angka akan semakin tinggi 0-9



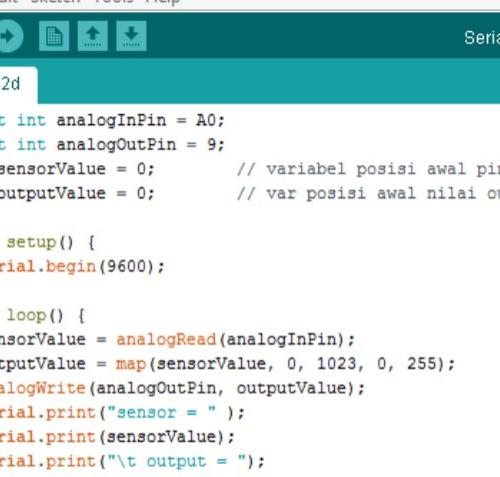
The image shows the Arduino IDE interface. The top window is titled "adc2c | Arduino 1.6.5" and contains the following sketch code:

```
void setup() {
  Serial.begin(9600); // setup port serial
}
void loop() {
  int sensorValue = analogRead(A0); // baca input analog pin 0
  int mappedSensorValue = map(sensorValue, 0, 1023, 0, 9);
  Serial.println(mappedSensorValue, DEC); // kirim ke port serial.
}
```

Below the IDE are two separate windows titled "COM23 (Arduino/Genuino Uno)". The left window shows the serial output starting with "0" repeated 15 times, indicating the potentiometer is at its minimum (left). The right window shows the serial output starting with "9" repeated 15 times, indicating the potentiometer is at its maximum (right). Both windows have "Autoscroll" checked and are set to "9600 baud".

#### 5. Percobaan 2d

Membaca tegangan adc di serial monitor jika potensiometer diputar ke kiri angka akan semakin kecil dan jika diputar ke kanan angka akan semakin tinggi 0-1023 dan 0-255



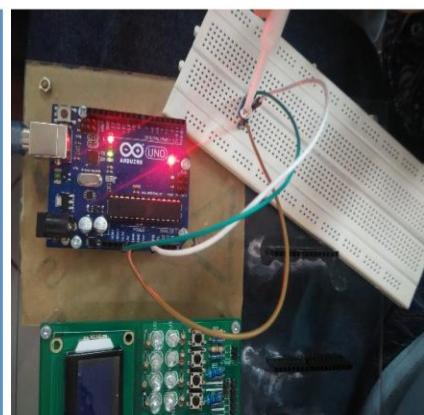
```
adc2d | Arduino 1.6.5
File Edit Sketch Tools Help
Serial Monitor
adc2d
const int analogInPin = A0;
const int analogOutPin = 9;
int sensorValue = 0;           // variabel posisi awal pin A0
int outputValue = 0;           // var posisi awal nilai output

void setup() {
  Serial.begin(9600);
}

void loop() {
  sensorValue = analogRead(analogInPin);
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(analogOutPin, outputValue);
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  delay(2);
}
```

```
COM3 (Arduino/Genuine Uno) - x
Send
sensor = 391    output = 97
sensor = 389    output = 96
sensor = 379    output = 94
sensor = 412    output = 102
sensor = 386    output = 96
sensor = 429    output = 106
sensor = 433    output = 107
sensor = 465    output = 115
sensor = 449    output = 111
sensor = 460    output = 114
sensor = 451    output = 112
sensor = 429    output = 106
sensor = 407    output = 101
sensor = 372    output = 92
sensor = 385    output = 95
sensor
```

```
sensor = 11      output = 2
sensor = 15      output = 3
sensor = 31      output = 7
sensor = 15      output = 3
sensor = 10      output = 2
sensor = 28      output = 6
sensor = 23      output = 5
sensor = 8       output = 1
sensor = 19      output = 4
sensor = 30      output = 7
sensor = 13      output = 3
sensor = 15      output = 3
sensor = 31      output = 7
sensor = 17      output = 4
sensor = 11      output = 2
sensor = 27      output = 6
sensor = 25      output = 6
```



G. LATIHAN

1. Buatlah eksperimen ADC dengan menggunakan komunikasi serial untuk menampilkan data di serial monitor dan di LED 4-bit:
    - dari nilai 0 = 0000 hingga nilai 255 = 1111

## PERCOBAAN 9

### KOMUNIKASI SERIAL

#### A. TUJUAN

1. Membuat aplikasi komunikasi serial antara PC dengan modul mikrokontroler arduino
2. Menampilkan dan mengamati data setelah pengiriman data dari arduino ke serial monitor PC

#### B. DASAR TEORI

Komunikasi serial adalah salah satu protokol komunikasi yang paling banyak diadopsi dan juga terdokumentasi dengan baik. Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi serial ada dua macam, asynchronous serial dan synchronous serial.

1. *Synchronous serial* adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan clock dan mengirimkan clock tersebut bersama-sama dengan data.

*Contoh : terdapat pada transmisi data keyboard.*

2. *Asynchronous serial* adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan clock namun hanya data yang ditransmisikan, tanpa clock.

*Contoh : UART pada serial port (COM) komputer.*

#### Konsep Baud Rate

Menghubungkan dua perangkat melalui komunikasi serial adalah kedua perangkat berkomunikasi dengan konfigurasi yang sama. Parameter komunikasi secara serial, diantaranya adalah baud rate, paket data, parity bit, dan synchronization bit. Baud rate : kecepatan data dikirim melalui

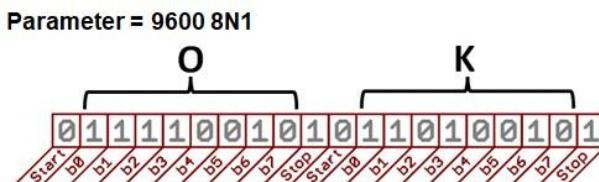
komunikasi serial. Satuan bit-per-second (bps). Standar kecepatan : 1200, 2400, 4800, 9600, 19200, 38400, 57600, dan 115200 bps.

### Contoh Pengiriman Data

Scenario pengiriman, yaitu 9600 8N1. Kode 9600 8N1 bermakna bahwa kecepatan yang digunakan 9600 baud, 8-bit data, tidak terdapat parity, dan 1-bit stop. Data akan dikirim dalam format ASCII (American Standard Code for Information Interchange). Dikirim kata OK (O dan K), maka komunikasi akan memiliki dua buah paket data.

- Karakter „O“ = 79d (0100 1111 b)
- Karakter „K“ = 75d (0100 1011b).

Kecepatan 9600 bit/detik, maka setiap bitnya memerlukan waktu selama  $1/9600 = 104$  mikrodetik/bit. Satu paket data untuk satu karakter terdiri dari 10 bit (8-bit data, 1-bit start dan 1-bit stop). Pengiriman satu karakter (yang terdiri dari 10-bit) akan membutuhkan waktu selama  $10 \times 104$  mikrodetik = 1.040 mikrodetik = 1,04 milidetik. Pengiriman kata „OK“ akan membutuhkan waktu sekitar 2 milidetik pada kecepatan 9600 bps.



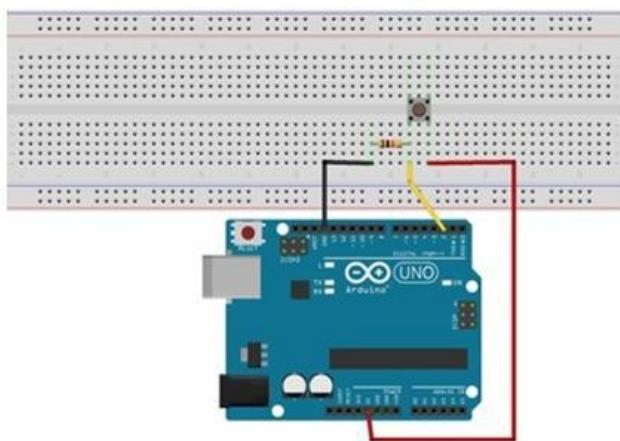
Gambar 1. Gambar Paket Serial “OK”

Setiap port komunikasi serial akan menggunakan sepasang pin digital. Satu pin untuk menerima, dan satu lagi untuk mengirim. Sebagai contoh, di Arduino UNO yang memiliki satu port komunikasi serial, pin #0 untuk menerima (RX), dan pin #1 untuk mengirim (TX). Semua board Arduino memiliki sedikitnya satu buah kontroler komunikasi serial. Hal ini menyebabkan komunikasi serial dengan board Arduino mudah untuk dilakukan.

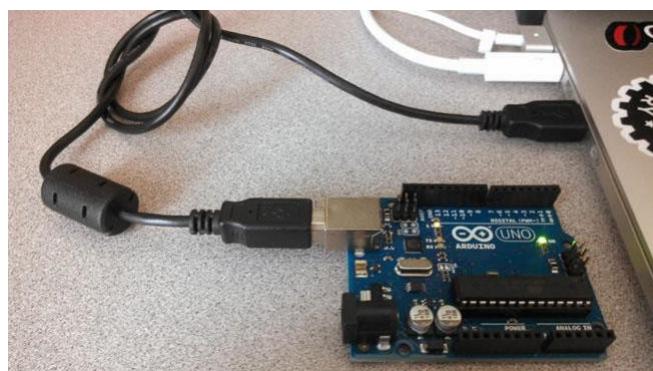
## C. HARDWARE

- 2 Buah Arduino Uno Board
  - Kabel jumper
  - Resistor  $10k\Omega$
  - Push Button

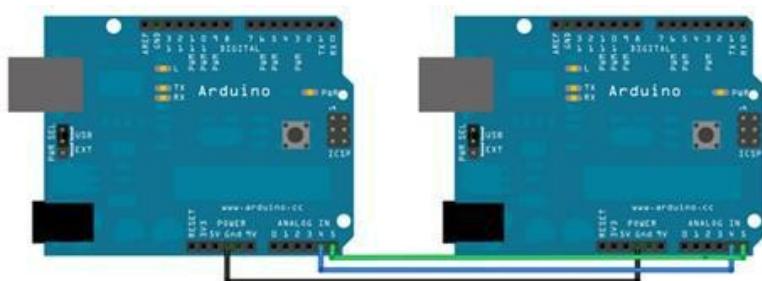
## D. RANGKAIAN



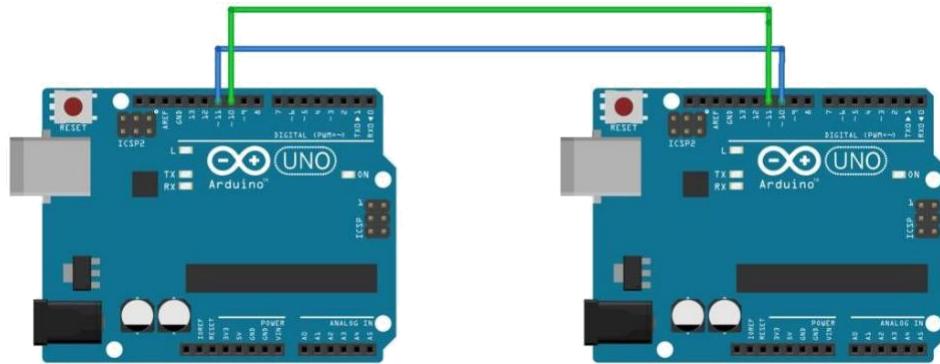
Gambar 2. Rangkaian Percobaan 1



Gambar 3. Rangkaian Percobaan 2



Gambar 4. Rangkaian Percobaan 3



Gambar 5. Rangkaian Latihan Eksperimen Arduino Chat

## E. LANGKAH PERCOBAAN

1. Tancapkan usb serial ke kemputer.
2. Pastikan drivernya sudah terinstall dan dikenali dengan benar.
3. Cek driver, lihat port COM dan samakan pada software arduino



4. Jalankan arduino dan buat programnya
5. Buka jendela komunikasi serial pada IDE arduino caranya, pilih Tools > Serial Monitor.

## F. PROGRAM PERCOBAAN

1. Percobaan 1, menampilkan nilai kondisi apabila button ditekan maka akan muncul 1 di serial monitor, apabila tidak maka muncul 0.

```

_1.s
int pushButton = 2;
void setup()
{
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}
void loop()
{
  int kondisiButton = digitalRead(pushButton);
  Serial.println(kondisiButton);
  delay(1000);
}

```

2. Percobaan 2, Menampilkan kata “Hello World!” pada Serial Monitor.



```

_2 §
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println("Hello World!");
  delay(1000);
}
  
```

3. Percobaan 3, Membuat program tx da rx dengan arduino



```

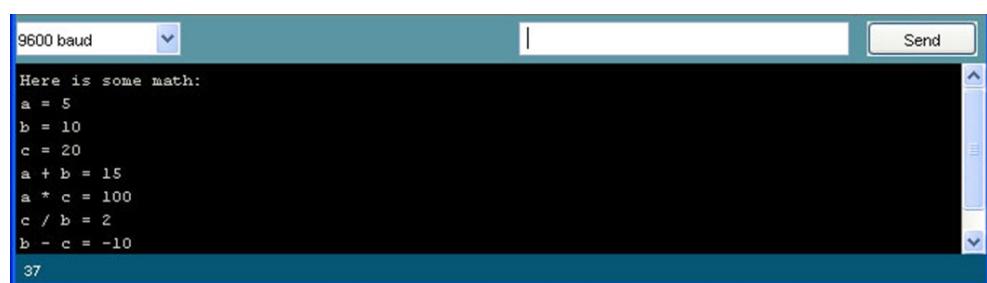
Pengirim_Tx_S
#include<Wire.h>
void setup()
{
  Wire.begin(2);
  Wire.onRequest(requestEvent);
}
void loop()
{
  delay(1000);
}

Void requestEvent()
{
  Wire.write("test");
  //for (int thisChar = 65; this Char < 75; thisChar++)
  //  /Wire.write(thisChar);
}

Penerima_Rx_S
#include<Wire.h>
void setup(){
  Wire.begin(2);
  Serial.begin(9600);
}
void loop(){
  Wire.requestFrom(2,6);
  while(Wire.available()){
    char c=Wire.read();
    Serial.print(c);
  }
  delay(500);
}
  
```

## G. LATIHAN

1. Buatlah eksperimen dengan menggunakan komunikasi serial untuk menampilkan hasil dari ekspresi matematika



9600 baud

Here is some math:  
a = 5  
b = 10  
c = 20  
a + b = 15  
a \* c = 100  
c / b = 2  
b - c = -10

2. Buatlah eksperimen dengan menggunakan komunikasi serial antara 2 modul arduino yang dapat berkomunikasi seperti chatting. Rangkaian terdapat pada Gambar 5.

## PRAKTIKUM 10

### APLIKASI LM 35

#### A. TUJUAN

Setelah melakukan percobaan ini, mahasiswa dapat:

- Membuat pendekripsi suhu menggunakan sensor LM35 dengan pemrograman arduino
- Mengubah data analog sensor LM35 menjadi data digital pada arduino

#### B. DASAR TEORI

Sensor adalah alat yang mengubah suatu besaran menjadi energi listrik. Sensor suhu (temperatur) LM35 ini dapat memberikan output 8-bit data yang menyatakan kondisi perubahan dari suhu maka akan terjadi perubahan data output yang dihasilkan, dimana perubahan tersebut berupa perbedaan tegangan yang dihasilkan. LM35 sebagai alat deteksi temperatur memiliki karakteristik sebagai berikut:

1. Bekerja pada rating tegangan 4V s/d 30V
2. Pembacaan temperatur berkisar antara  $-55^{\circ}\text{C}$  s/d  $150^{\circ}\text{C}$
3. Dengan kenaikan temperatur  $1^{\circ}\text{C}$  maka tegangan output akan naik sebesar 10mV
4. Memiliki arus drain kurang dari 60  $\mu\text{A}$

Mendeteksi suhu 0-100 derajat Celcius dengan karakteristik 10mV pada output mewakili 1 derajat Celsius. Tegangan output  $300\text{mV} = 30$  derajat Celcius, tegangan output  $230\text{mV} = 23$  derajat Celcius

#### Menghitung nilai temperatur:

**Cara 1:** Suhu = input analog x 0.488

**Cara 2:** Suhu =  $(5.0 * \text{analogRead(tempPin)} * 100.0) / 1024;$

**Cara 3:**  $1^{\circ}\text{C} = 10\text{ mV} / 4.88\text{ mV} = 2.0491$  maka

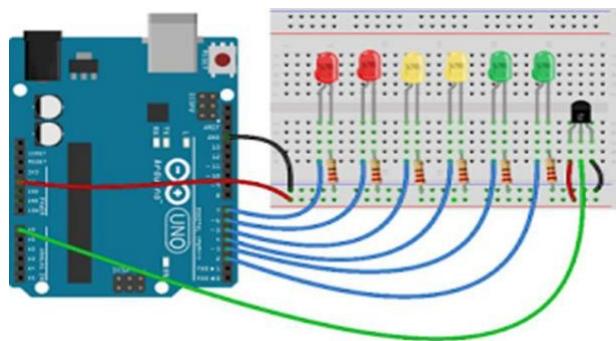
$$\text{Suhu} = \text{input analog} / 2.0491$$

#### C. PERALATAN

- Arduino Uno Board
- 1x Breadboard

- 1x LM35 Sensor Suhu
- 2x LED hijau
- 2x LED kuning
- 2x LED merah
- 6x Resistor 220ohm
- Kabel Jumper

## RANGKAIAN



- Pasang dari GND dan 5V Arduino ke Breadboard.
- Pasang kaki kiri LM35 ke 5V, kaki kanan LM35 ke GND, kaki tengah LM35 ke pin A0 Arduino.
- Pasang kaki Positif LED hijau ke pin 2 dan pin 3, LED kuning ke pin 4 dan pin 5, LED merah ke pin 6 dan pin 7 Arduino.
- Setiap kaki negatif LED dipasang ke GND menggunakan Resistor 220 ohm.

## E. LANGKAH-LANGKAH

- Tancapkan usb serial ke komputer.
- Pastikan drivernya sudah terinstall dan dikenali dengan benar.
- Cek driver, lihat port COM dan samakan pada software arduino



- Jalankan arduino dan buat programnya
- Buka jendela komunikasi serial pada IDE arduino caranya, pilih Tools > Serial Monitor.

## F. LISTING PROGRAM

1. Membaca temperature ruangan dalam satuan celcius dengan rumus perhitungan:  $\text{DataInputLM35} * 0.488$ . Temperatur ditampilkan pada serial monitor dengan delay 500mS.

```
perc_LM_1 §
int LM35 = A1; // variabel LM35 untuk pin A1 Arduino
int nilaiLM35= 0; // variabel nilaiLM35 = menyimpan nilai sensor
void setup(){Serial.begin(9600);
}
void loop(){nilaiLM35 = analogRead(LM35); // simpan nilai LM35 ke nilaiLM35
nilaiLM35 = nilaiLM35 * 0.488; // konversi nilai dari LM35 = 0 Celcius
Serial.println(nilaiLM35);
delay(500);
}
```

COM8 (Arduino/Genuino Uno)

```
27
27
27
27
27
27
27
```

2. Membaca temperature ruangan dalam satuan celcius dengan rumus perhitungan  $(\text{DataInputLM35}/1024.0) * 5000$ . Temperatur ditampilkan pada serial monitor dengan delay 500mS.

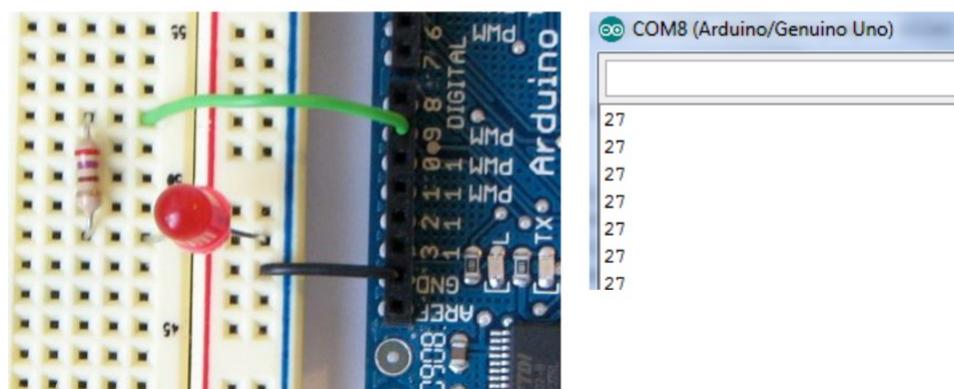
```
perc_LM_2 §
int val;
int tempPin = A1;
void setup(){Serial.begin(9600);
}
void loop(){val = analogRead(tempPin);
float mv = ( val/1024.0)*5000;
float cel = mv/10;
float fahr = (cel*9)/5 + 32;
Serial.print("TEMPRATURE = ");
Serial.print(cel);
Serial.print("°C");
Serial.println();
delay(1000);
Serial.print("TEMPRATURE = ");
Serial.print(fahr);
Serial.print("°F");
Serial.println();
}
```

COM5

```
Suhu: 29.00
```

3. Membaca temperature ruangan dalam satuan celcius dengan rumus perhitungan  $\text{DataInputLM35} * 0.488$ . Dengan tambahan apabila temperature ruangan yang dibaca oleh LM35 sama dengan 28 derajat celcius maka ketiga LED akan menyala. Temperatur ditampilkan pada serial monitor dengan delay 500mS.

```
perc_LM_3 §
int LM35 = A0; int nilaiLM35= 0;
int LED1= 2; // membuat variabel LED1 untuk Pin 2 digital
int LED2= 3; // membuat variabel LED2 untuk Pin 3 digital
int LED3= 4; // membuat variabel LED3 untuk Pin 4 digital
void setup(){
Serial.begin(9600);
pinMode(LED1, OUTPUT); // LED1 = OUTPUT
pinMode(LED2, OUTPUT); // LED2 = OUTPUT
pinMode(LED3, OUTPUT); // LED3 = OUTPUT
}
void loop(){
nilaiLM35 = analogRead(LM35);
nilaiLM35 = nilaiLM35 * 0.488;
Serial.println(nilaiLM35);
delay(500);
if (nilaiLM35 > 28) // jika nilaiLM35 sama dengan 28
{
  digitalWrite(LED1, HIGH); // LED1 menyala
  digitalWrite(LED2, HIGH); // LED2 menyala
  digitalWrite(LED3, HIGH); // LED3 menyala
} else {
  digitalWrite(LED1, LOW); // LED1 mati
  digitalWrite(LED2, LOW); // LED2 mati
  digitalWrite(LED3, LOW); // LED3 mati
}
}
```



## **G. LATIHAN**

1. Buatlah program, "Jika nilai LM35 = 28, maka semua LED warna hijau menyala. Jika nilai LM35 = 29, maka semua LED warna kuning menyala. Dan jika nilai LM35 = 30, maka semua LED warna merah menyala."
2. Buatlah program untuk membaca data input analog dan mengkonversikan ke derajat Celcius menggunakan rumus cara-3. Data input analog dan hasil konversinya ditampilkan pada serial monitor.

## PERCOBAAN 11

### ANTARMUKA MIKROKONTROLER DENGAN LDR

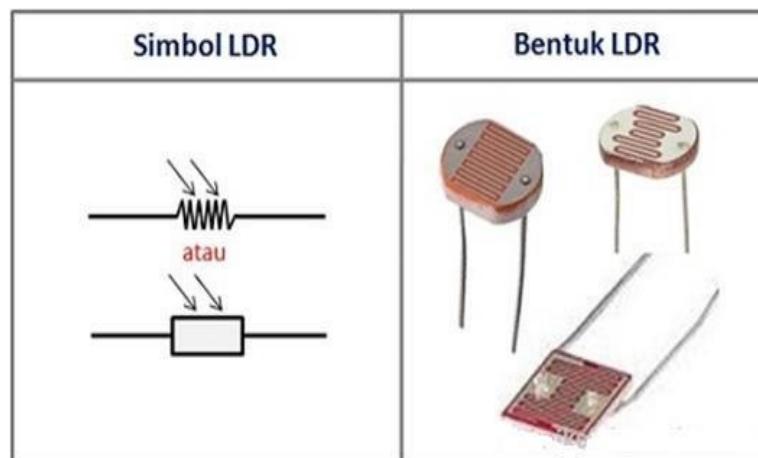
#### A. TUJUAN

1. Membuat pendekripsi Cahaya menggunakan sensor LDR dengan pemrograman Arduino.
2. Mengubah data analog sensor LDR menjadi data digital pada Arduino.

#### B. DASAR TEORI

##### Pengertian LDR

*Light Dependent Resistor* (LDR) adalah jenis resistor yang nilai hambatan atau resistansinya tergantung pada intensitas cahaya yang diterimanya. Nilai Hambatan LDR akan menurun pada saat cahaya terang dan nilai Hambatannya akan menjadi tinggi jika dalam kondisi gelap. Nilai Hambatan *LDR* akan mencapai  $200\text{ k}\Omega$  pada kondisi gelap dan menurun menjadi  $500\text{ }\Omega$  pada kondisi cahaya terang.



Gambar 1. Bentuk dan Simbol LDR

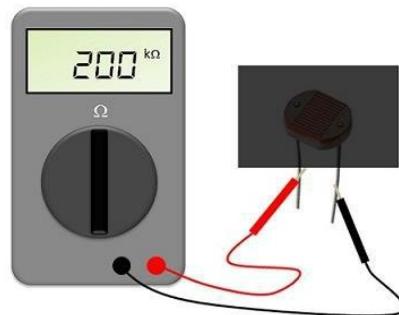
##### Cara mengukur LDR

1. Ketika LDR di beri cahaya, maka cara pengukurannya adalah seperti pada gambar dibawah ini.



Gambar 2. Cara pengukuran ketika LDR di beri cahaya

2. Ketika LDR tidak di beri cahaya atau LDR di tutup dengan bahan yang tidak tembus cahaya, maka cara pengukurannya adalah



Gambar 3. Cara pengukuran ketika LDR tidak di beri cahaya

### Fungsi LDR

1. Menghantarkan arus listrik jika intensitas cahaya kondisi terang dan menghambat arus listrik dalam kondisi gelap.
2. Sebagai saklar otomatis berdasarkan cahaya.
3. Aplikasi: sensor pada lampu penerang jalan, lampu kamar tidur, rangkaian anti maling, shutter kamera, alarm dan lain sebagainya.

### Sensor Cahaya

Selain LDR adapula jenis sensor cahaya yang lain, sensor cahaya sendiri merupakan komponen elektronika yang dapat memberikan perubahan besaran elektrik pada saat terjadi perubahan intensitas cahaya yang diterima, seperti yang dipakai pada penerima remote televisi dan pada lampu penerangan jalan otomatis.

## Jenis-Jenis Sensor Cahaya

Berdasarkan perubahan output ada 2 yaitu :

1. Sensor cahaya tipe fotovoltaik
2. Sensor cahaya tipe fotokonduktif

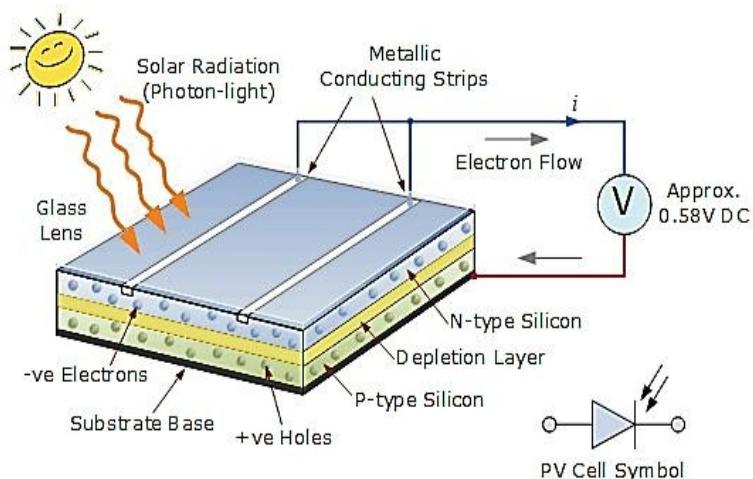
Berdasarkan cahaya yang diterima :

1. Sensor cahaya infra merah
2. Sensor cahaya ultraviolet

Berdasarkan perubahan Output

1. Tipe Fotovoltaik

Sensor cahaya yang dapat memberikan perubahan tegangan pada outputnya dan juga mengubah energi sinar langsung menjadi energi listrik. Contoh: solar cell atau sel surya.



Gambar 4. Contoh Cara Kerja Sensor Cahaya Solar Cell

2. Tipe fotokonduktif

Sensor cahaya yang memberikan perubahan resistansi pada terminal outputnya sesuai dengan perubahan intensitas cahaya yang diterimanya.



Gambar 5. Bentuk Fisik LDR (kiri), Photo Transistor (tengah), Photo Diode (Kanan)

Berdasarkan cahaya yang diterima

1. Sensor Cahaya Infra Merah

Sensor cahaya yang hanya akan merespon perubahan cahaya infra merah. Contoh: photo transistor atau photo dioda. Jika menerima penceran cahaya infra merah maka pada terminal outputnya akan memberikan perubahan resistansi

2. Sensor Cahaya Ultraviolet

Sensor cahaya yang hanya merespon perubahan intensitas cahaya ultraviolet yang mengenainya. Contoh: Modul sensor cahaya Uvtron Memberikan perubahan besaran listrik (tegangan) pada terminal outputnya pada saat menerima perubahan intensitas penceran cahaya ultraviolet.

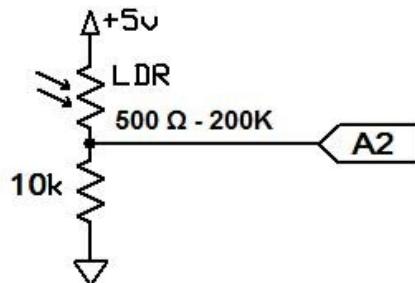


Gambar 6. Bentuk Fisik Modul Sensor Cahaya Uvtron

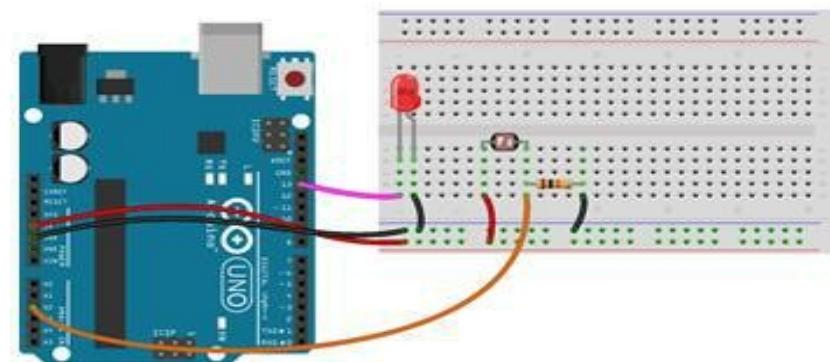
### C. HARDWARE

- Arduino Uno Board
- 1x Breadboard
- 1x Sensor cahaya LDR
- 1x Resistor 10k
- Kabel jumper

### D. RANGKAIAN



Gambar 7. Skema Rangkaian Sensor LDR dengan Pin A2 (Analog) Arduino



Gambar 8. Rangkaian Dasar Implementasi Sensor LDR Menggunakan Arduino Yang Digunakan Sebagai Percobaan

### E. LANGKAH PERCOBAAN

1. Hubungkan 5V dan GND dari Arduino ke Breadboard.
2. Hubungkan kaki kiri LDR ke 5V.
3. Hubungkan kaki kanan LDR ke pin A2 Arduino.
4. Hubungkan kaki kiri resistor ke celah antara kaki kanan LDR dan GND arduino.

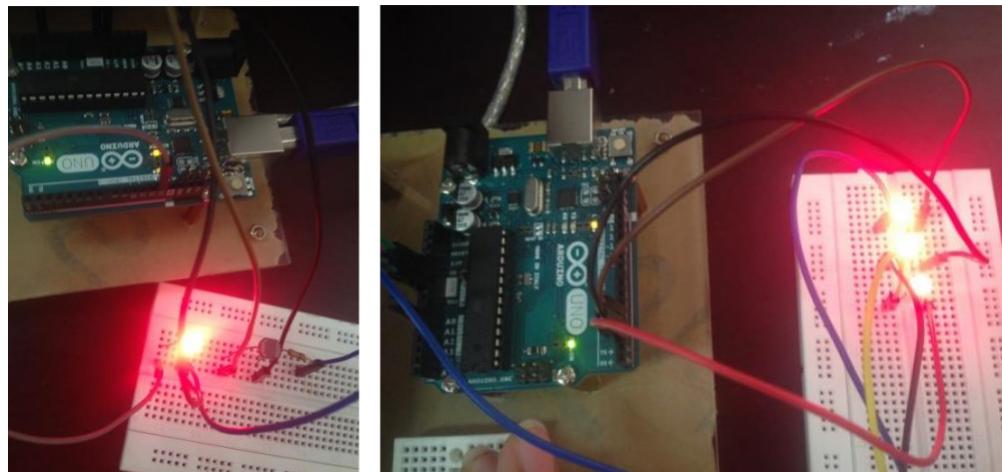
5. Hubungkan kaki kanan resistor ke GND.
6. Hubungkan LED dengan pin 13 Arduino.

## F. PROGRAM

Program dibawah merupakan program untuk membaca intesitas cahaya menggunakan sensor LDR dan ditampilkan pada serial monitor, apabila nilai sensor LDR yang dibaca oleh arduino bernilai kurang dari 500 maka LED menyala, apabila lebih dari 500 maka LED padam.

```
perc_LDR
int LDR=A2;
int LED=2;
int nilaiLDR=0;
void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  nilaiLDR=analogRead(LDR);
  Serial.print("Nilai LDR =");
  Serial.println(nilaiLDR);
  if(nilaiLDR < 500){
    digitalWrite(LED,HIGH);
  } else if(nilaiLDR > 500) {
    digitalWrite(LED,LOW);
  }
}
```

Hasil Program :



## **G. LATIHAN**

1. Program LED BLINK, dimana kecepatan BLINKnya diatur melalui nilai dari Sensor LDR.
2. Buatlah program untuk membuat lampu LED menyala terang dan redup berdasarkan nilai dari sensor LDR.

## PRAKTIKUM 12

### ANTARMUKA MIKROKONTROLER DENGAN LAMPU AC 220V

#### A. TUJUAN

Setelah melakukan percobaan ini, mahasiswa dapat:

- Mengendalikan nyala lampu AC 220 Volt (On dan Off) menggunakan driver
- Membuat rangkaian driver Transistor dan Relay untuk mengendalikan lampu AC 220 Volt

#### B. DASAR TEORI

##### **RELAY:**

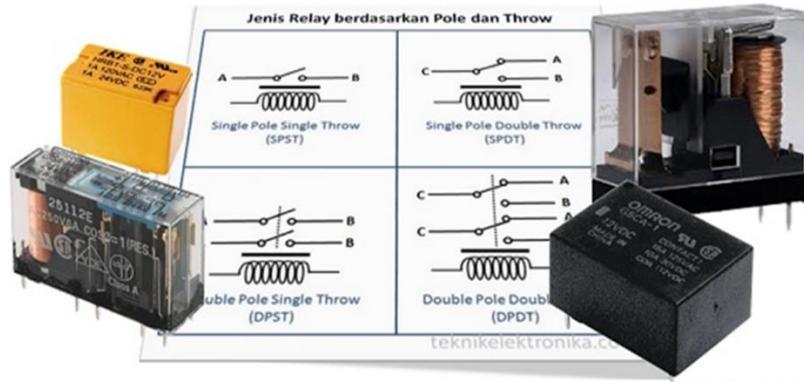
Sebagai komponen yang dapat mengimplementasikan logika switching. Merupakan “otak” dari rangkaian pengendali. relay yang paling sederhana ialah relay elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. Secara sederhana relay elektromekanis didefinisikan sebagai berikut :

- Alat yang menggunakan gaya elektromagnetik untuk menutup atau membuka kontak saklar.
- Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

**Jenis relay:** dari desain saklar relay maka relay dibedakan menjadi :

- Single Pole Single Throw (SPST), relay ini memiliki 4 terminal yaitu 2 terminal untuk input kumapanan elektromagnet dan 2 terminal saklar. Relay ini hanya memiliki posisi NO (Normally Open) saja.
- Single Pole Double Throw (SPDT), relay ini memiliki 5 terminal yaitu terdiri dari 2 terminal untuk input kumapanan elektromagnetik dan 3 terminal saklar. relay jenis ini memiliki 2 kondisi NO dan NC.
- Double Pole Single Throw (DPST), relay jenis ini memiliki 6 terminal yaitu terdiri dari 2 terminal untuk input kumapanan elektromagnetik dan 4 terminal saklar untuk 2 saklar yang masing-masing saklar hanya memiliki kondisi NO saja.

- Double Pole Double Throw (DPDT), relay jenis ini memiliki 8 terminal yang terdiri dari 2 terminal untuk kumparan elektromagnetik dan 6 terminal untuk 2 saklar dengan 2 kondisi NC dan NO untuk masing-masing saklarnya.



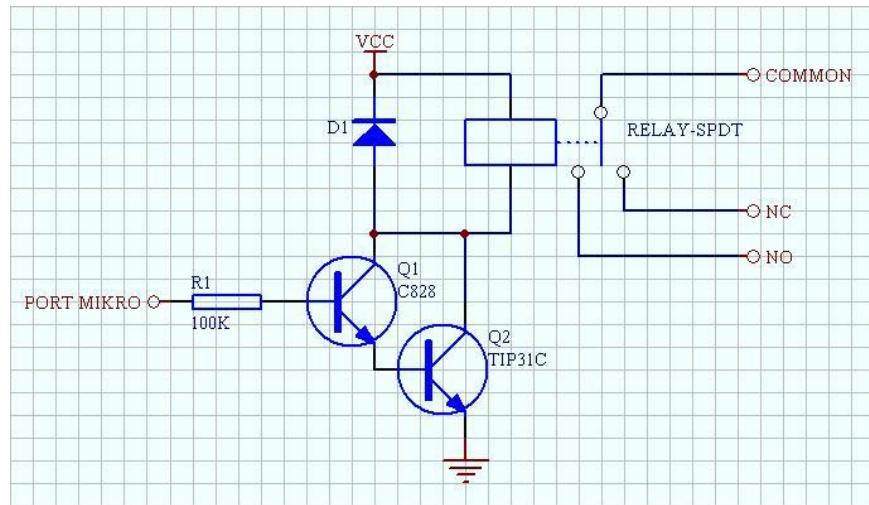
Gambar 1. Jenis Relay

### RANGKAIAN DRIVER:

Relay SPDT 12V dengan kapasitas arus 5A. Dari hasil pengukuran nilai resistansi kumparan relay adalah sebesar 358 ohm. Arus yang dibutuhkan adalah sebesar  $12V / 358 \text{ Ohm} = 33,5 \text{ mA}$ . Sehingga transistor harus dapat menghasilkan arus sedikitnya 2-3 kali lebih besar dari 33,5 mA, yakni sekitar 100 mA. Transistor yang digunakan adalah 2 buah transistor NPN tipe C828 memiliki penguatan arus DC (hfe) sekitar 130 – 520 kali. Jika penguatan arusnya sebesar 100 kali. Transistor C828 memiliki  $V_{BE} = 0,8V$ .

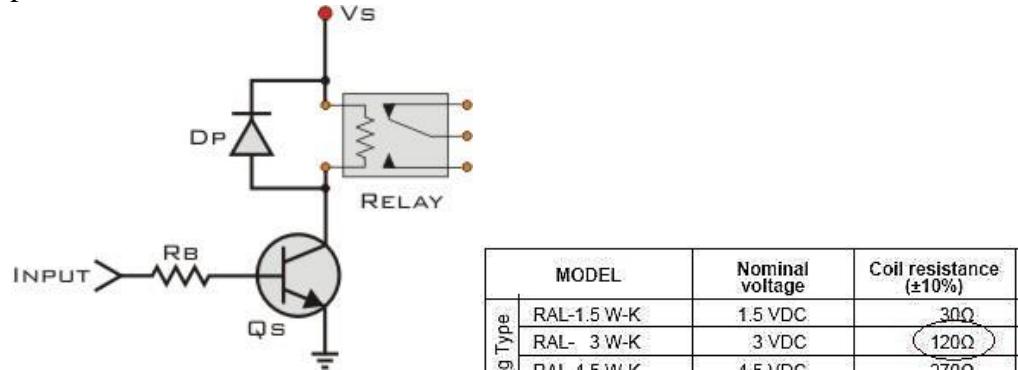
Transistor disusun secara Darlington sehingga penguatan arusnya menjadi  $100 \times 100 = 10.000$  kali. Arus basis minimal :  $I_B = 100 / 10000 = \pm 10 \mu\text{A}$ . Jika  $V_{BE}$  bernilai 0,8 volt dan tegangan keluaran logika 1 mikrokontroler bernilai 4,8 volt, maka  $R_B = (4,8 - 0,8 - 0,8) / 10E-6 = 320000 \text{ ohm}$ . Pemasangan diode 1N4002 berfungsi mencegah arus transien yang ditimbulkan oleh kumparan relay.

Jadi rangkaian di atas sangat cocok digunakan pada aplikasi menggunakan mikrokontroler karena arus source port I/O mikrokontroler biasanya hanya 20mA.



Gambar 2. Rangkaian Driver

Sebuah rangkaian relay dengan spesifikasi komponen relay seperti diperlihatkan pada tabel.



Relay type RAL3W-K yang Coil Resistance-nya adalah 120 Ohm. Maka, arus beban (IL) dapat dihitung sebagai berikut :

$$IL = VS/RL$$

IL = arus beban

VS = tegangan catu

RL = resistansi beban (coil resistance)

$$IL = 3/120 = 25 \text{ mA}$$

Dioda DP diparalel dengan coil relay, tujuannya untuk memproteksi arus balik ke transistor jika coil dalam keadaan OFF. Dioda yang dapat digunakan antara lain IN4001 atau IN4004

hFE minimum dari transistor Qs adalah :

“5 kali arus beban dibagi arus input pada basis transistor”

$$hFE(\min) = 5 \times IL / \text{arus input}$$

Misalnya input basis transistor Qs adalah output digital Xbe, 4mA, maka :

$$hFE(\min) = 5 \times 25/4 = 31,25$$

Jadi, transistor Qs yg dipilih hFE nya harus lebih besar dari 31,25. Misalnya 2N2222, BC547, BC 337 yang hFE nya sekitar 100.

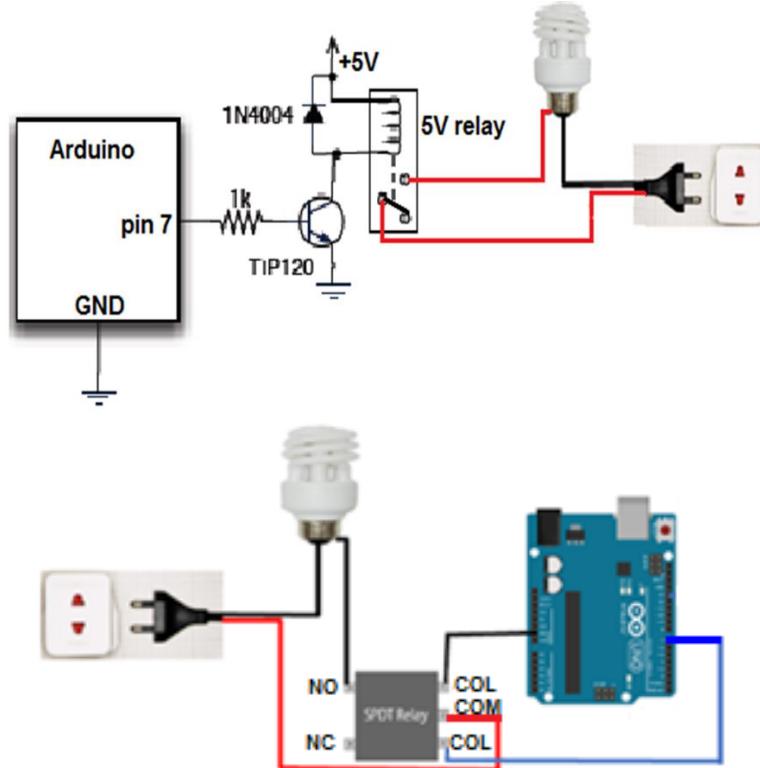
Nilai RB (Resistor Basis) adalah :

$RB = (VSxhFE)/(5xIL) = (3 \times 100)/(5 \times 0,025) = 750 \text{ Ohm} \rightarrow \text{nilai yg mendekati adalah } RB = 680 \text{ Ohm.}$

### C. PERALATAN DAN RANGKAIAN

- Arduino Uno Board
- 1x Breadboard
- 1x Lampu AC 220 V
- 1x Relay
- 1x Transistor
- 1x Resistor
- Kabel Jumper

Rangkaian :



Gambar 3. Rangkaian Percobaan

## D. LANGKAH-LANGKAH

- Tancapkan usb serial ke kemputer.
- Pastikan drivernya sudah terinstall dan dikenali dengan benar.
- Cek driver, lihat port COM dan samakan pada software arduino



- Buat rangkaian seperti pada gambar 3.
- Jalankan arduino dan buat programnya

## F. LISTING PROGRAM

1. Program kontrol on-off lampu AC 220 volt dengan delay 2 detik

```
int lampu=7;
void setup(){
  pinMode(lampu, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  digitalWrite(lampu, HIGH); // Turns ON Relays
  Serial.println("Light ON");
  delay(2000);
  digitalWrite(lampu, LOW); // Turns Relay Off
  Serial.println("Light OFF");
  delay(2000);
}
```

2. Program kontrol Lampu AC 220 volt menggunakan tombol keyboard a dan b.

```

int lampu = 7;
int out;
void setup()
{
  Serial.begin(9600);
  pinMode(lampu, OUTPUT);
}
void loop()
{
  if (Serial.available()>0)
  {
    int baca = Serial.read();
    if (baca =='a')
    {
      out=1;
    }
    if (baca =='b')
    {
      out=0;
    }
    if(out==1)
    {
      digitalWrite(lampu,HIGH);
      Serial.println("LAMPU
NYALA");
    }
    else if(out==0)
    {
      digitalWrite(lampu,LOW);
      Serial.println("LAMPU
MATI");
    }
  }
}

```

## G. LATIHAN

1. Buatlah aplikasi kontrol Lampu AC 220V dengan ketentuan :

- Saat angka 1 dikirim ke arduino, maka Lampu-1 menyala. Sedangkan saat angka 0 dikirim ke arduino maka Lampu akan padam.
- Saat angka 2 dikirim ke arduino, maka Lampu-2 menyala. Sedangkan saat angka 0 dikirim ke arduino maka Lampu-2 akan padam.
- Saat angka 3 dikirim ke arduino, maka Lampu-1 dan Lampu-2 menyala. Sedangkan saat angka 0 dikirim ke arduino maka Lampu-1 & Lampu-2 akan padam.

## PERCOBAAN 13

### ANTARMUKA MIKROKONTROLER DENGAN MOTOR DC

#### A. TUJUAN

1. Membuat kontrol Motor DC Dengan Menggunakan PWM
2. Membuat program untuk kontrol arah putaran dan kecepatan Motor DC

#### B. DASAR TEORI

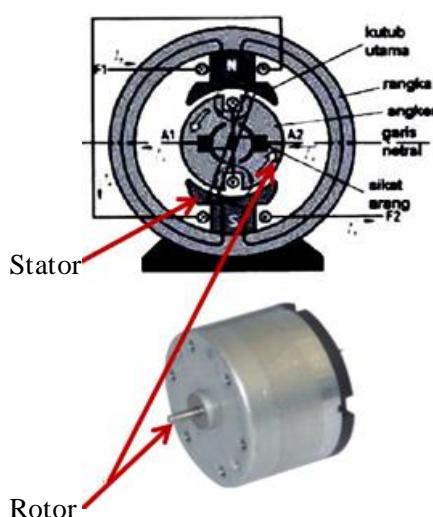
*MOTOR LISTRIK* merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor ditinjau dari catu nya dibagi dua jenis, yaitu motor arus searah (Motor DC) dan motor arus bolak-balik (Motor AC).

##### Motor DC

Definisi MOTOR DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya.

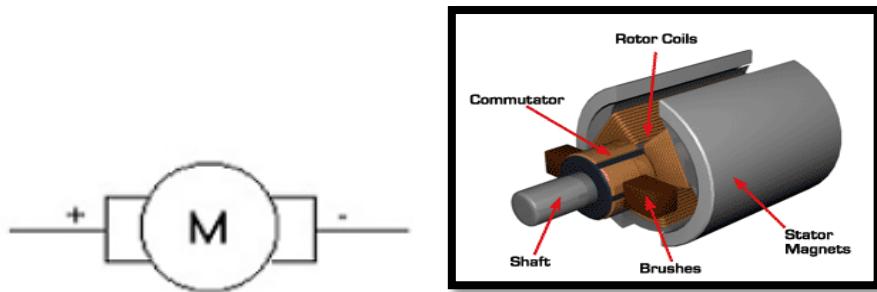
Memiliki 2 bagian dasar :

1. Bagian yang tetap/stasioner = stator. Stator ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen.
2. Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir.



Gambar 1. Motor DC

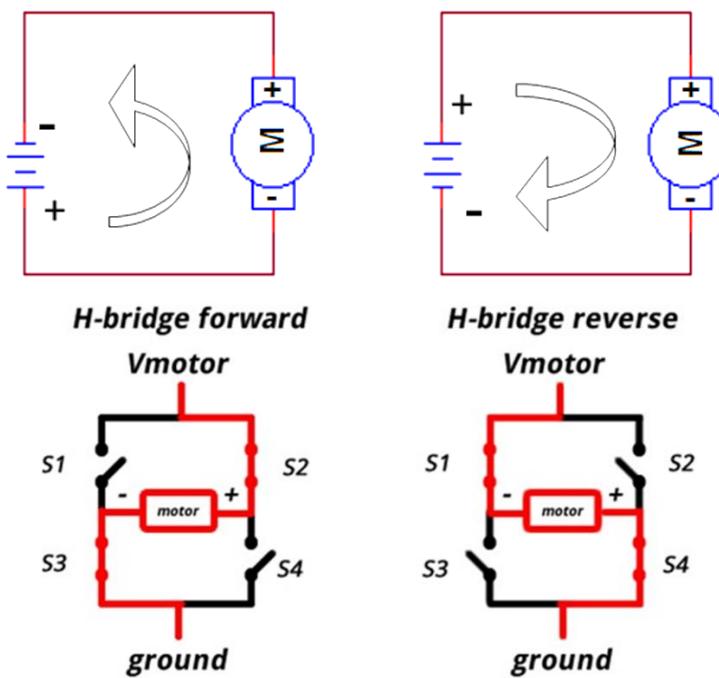
## Simbol Motor DC



Gambar 2. Simbol dan bagian motor DC

## Pengaturan Arah Putar Motor DC

- Mengubah arah arus yang mengalir melalui motor tersebut atau dengan mengubah polaritas tegangan.
  - Aplikasi pengaturan arah putaran dengan mikrokontroler menggunakan rangkaian driver dapat berupa IC, Transistor, H-Bridge atau dengan Relay.
- Sebagai contoh :



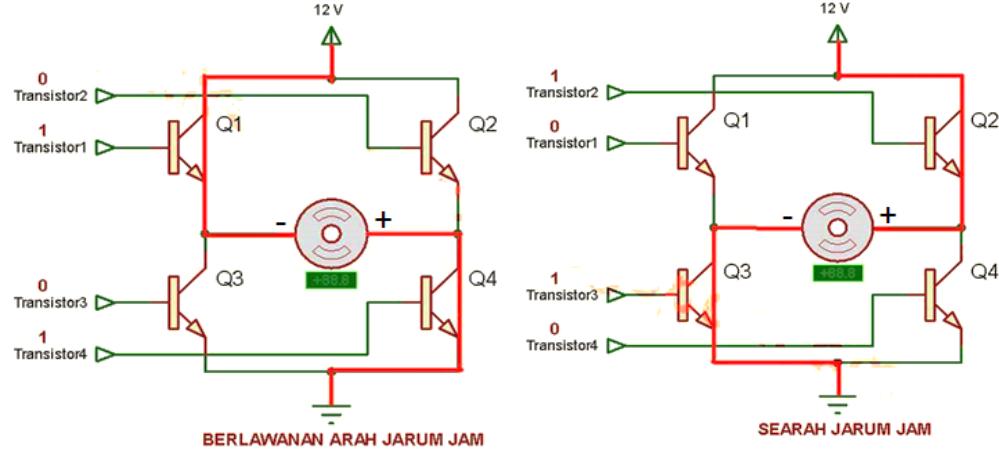
Gambar 3. Cara pengaturan arah putaran motor DC

## Driver H-Bridge

Rangkaian H-Bridge adalah rangkaian digunakan untuk mengendalikan motor DC. Rangkaian ini terdiri dari 4 buah transistor yang masing-masing bisa diaktifkan, dan sebuah motor DC di bagian tengah.

Prinsip kerja rangkaian ini yaitu apabila Q1 dan Q4 aktif maka motor akan bergerak ke suatu arah, karena ada arus yang mengalir dari Vcc ke ground. Sedangkan apabila Q2 dan Q3 aktif maka motor akan bergerak ke arah yang berlawanan, karena arus yang mengalir menuju motor arahnya berlawanan dibanding sebelumnya. Disini kondisi Q2 Q3 aktif tidak boleh bersamaan dengan kondisi Q4 Q1 aktif, karena bisa menyebabkan rangkaian dan motor yang rusak.

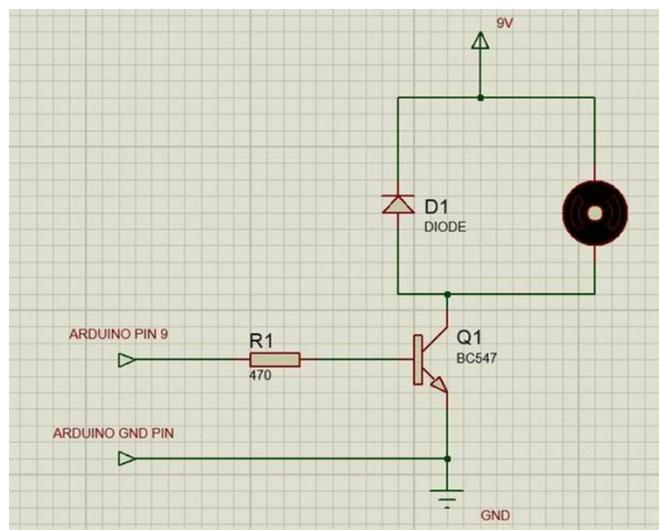
Hal ini bisa dicegah dengan mengantisipasinya pada program arduino.



Gambar 4. Driver H-Bridge

### Pengaturan Kecepatan Putar Motor DC

Pengaturan kecepatan putar motor DC ditunjukkan pada gambar berikut

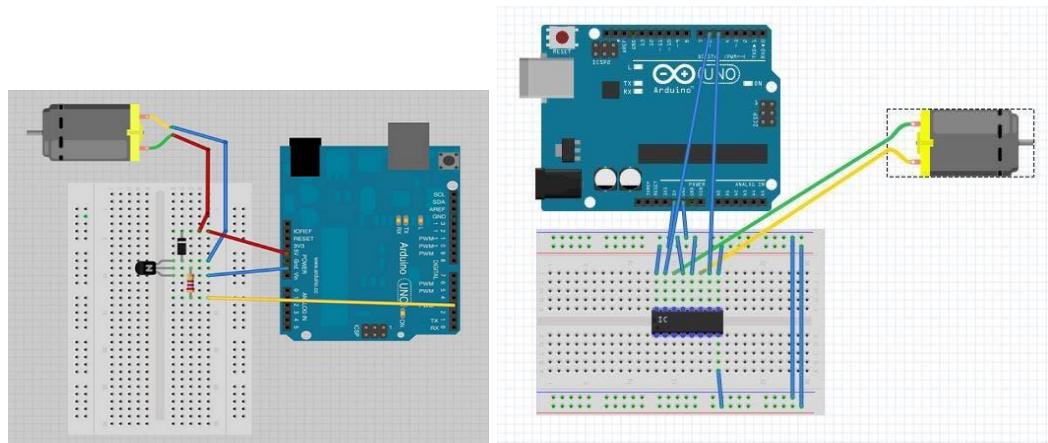


Gambar 5. Rangkaian pengaturan kecepatan putar motor DC

### C. HARDWARE

- Arduino Uno Board
- 1x Breadboard
- 1x motor DC
- 1x Resistor, transistor dan dioda
- Kabel jumper

### D. RANGKAIAN



Gambar 6. Rangkaian percobaan

### E. LANGKAH PERCOBAAN

1. Buat rangkaian seperti gambar 6.
2. Aktifkan software IDE arduino.
3. Setting Port COM IDE Arduino dan komputer.
4. Tulis program percobaan ke IDE Arduino
5. Upload dan jalankan program tersebut
6. Amati apakah hasil dari program percobaan sesuai yang diharapkan

### F. PROGRAM PERCOBAAN

#### 1. Menggunakan driver Transistor

```
int motorPin = 3;  
void setup() {  
}  
void loop() {  
    digitalWrite(motorPin, HIGH);  
}
```

```

int motorPin = 9;
void setup() {
    pinMode(motorPin, OUTPUT);
    Serial.begin(9600);
    while (! Serial);
    Serial.println("Speed 0 to 255");
}
void loop() {
    if (Serial.available()) {
        int speed = Serial.parseInt();
        if (speed >= 0 && speed <= 255) {
            analogWrite(motorPin, speed);
        }
    }
}

```

## 2. Menggunakan driver IC L293D

```

void setup() {
}
void loop() {
    for(int i = 0; i<255; i++){
        analogWrite(5, i);
        analogWrite(6, 0);
        delay(5);
    }
    for(int i = 0; i<255; i++){
        analogWrite(5, 255-i);
        analogWrite(6, 0);
        delay(5);
    }
}

```

```

for(int i = 0; i<255; i++){
    analogWrite(5, 0);
    analogWrite(6, i);
    delay(5);
}
for(int i = 0; i<255; i++){
    analogWrite(5, 0);
    analogWrite(6, 255-i);
    delay(5);
}

```

## G. LATIHAN

1. Dengan menambahkan komponen satu saklar, buatlah kontrol arah putaran motor DC.
2. Dengan menambahkan komponen 3 saklar, buatlah kontrol kecepatan putaran motor DC (pelan, sedang dan cepat).

## PERCOBAAN 14

### ANTARMUKA MIKROKONTROLER DENGAN MOTOR DC

#### A. TUJUAN

- Membuat program untuk mendeteksi benda
- Mengukur Jarak benda terhadap sensor menggunakan arduino

#### B. DASAR TEORI

##### Sensor ultrasonik

Kerja dari sensor ini didasarkan prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi sebuah benda spesifik yang ada dalam frekuensinya. Sensor ultrasonik adalah sebuah sensor yang memanfaatkan pancaran gelombang ultrasonic. Gelombang ultrasonik merupakan gelombang akustik yang memiliki frekuensi mulai 20 kHz hingga sekitar 20 MHz. Selisih waktu ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut. Jika gelombang ultrasonik berjalan melalui sebuah medium, Secara matematis besarnya jarak dapat dihitung sebagai berikut:

$$s = v \cdot t/2$$

Dimana :

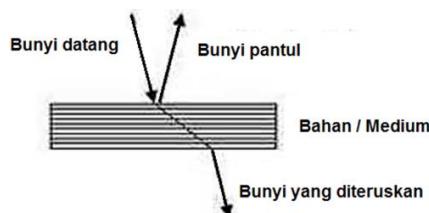
**S** adalah jarak dalam satuan meter,

**V** adalah kecepatan gelombang suara yaitu 340 m/detik dan

**t** adalah waktu tempuh dalam satuan detik.

##### Fenomena gelombang ultrasonik saat ada penghalang

Ketika gelombang ultrasonik menumbuk suatu penghalang maka sebagian gelombang tersebut akan dipantulkan sebagian diserap dan sebagian yang lain akan diteruskan.



Gambar 1. Gelombang ultrasonik saat ada penghalang

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran fisis (bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut piezoelektrik. Piezoelektrik ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 KHz sampai 400 KHz ketika sebuah osilator diterapkan pada benda tersebut.

Rangkaian penyusun sensor ultrasonik ini terdiri dari transmitter, receiver, dan komparator.

### **Bagian-bagian dari sensor ultrasonik**

#### **Piezoelektrik**

Mengubah energi listrik menjadi energi mekanik. Tegangan input yang digunakan menyebabkan bagian keramik meregang dan memancarkan gelombang ultrasonik. Tipe operasi transmisi elemen piezoelektrik sekitar frekuensi 32 kHz.

#### **Transmitter**

sebuah alat yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi sebesar 40 kHz yang dibangkitkan dari sebuah osilator.

#### **Receiver**

Receiver terdiri dari transduser ultrasonik menggunakan bahan piezoelektrik, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari transmitter yang dikenakan pada permukaan suatu benda atau gelombang langsung LOS (Line of Sight) dari transmitter.

### **Sensor Ultrasonic PING**

Sensor jarak ultrasonik ping adalah sensor dengan frekuensi 40 KHz. Sebagai sensor Jarak: menggunakan Sensor Ultrasonic HC-SR04 dengan pemrograman Arduino. Aplikasi untuk pengukuran tinggi air, tinggi badan, sebagai sensor navigasi untuk robot, dan lain lain.

Memiliki kemampuan deteksi yang sangat baik dengan akurasi tinggi dan pembacaan stabil. Kemampuan mengukur jarak 2cm - 400cm atau 1 inci sampai 4 meter serta tidak terpengaruh oleh sinar matahari.

## Bentuk sensor ultrasonic



Gambar 2. Sensor ultrasonic

### HC-SR04

1. **VCC** = 5V Power Supply. Pin sumber tegangan positif sensor.
2. **Trig** = Trigger/Penyulut. Pin ini yang digunakan untuk membangkitkan sinyal ultrasonik.
3. **Echo** = Receive/Indikator. Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik.
4. **GND** = Ground/0V Power Supply. Pin sumber tegangan negatif sensor.

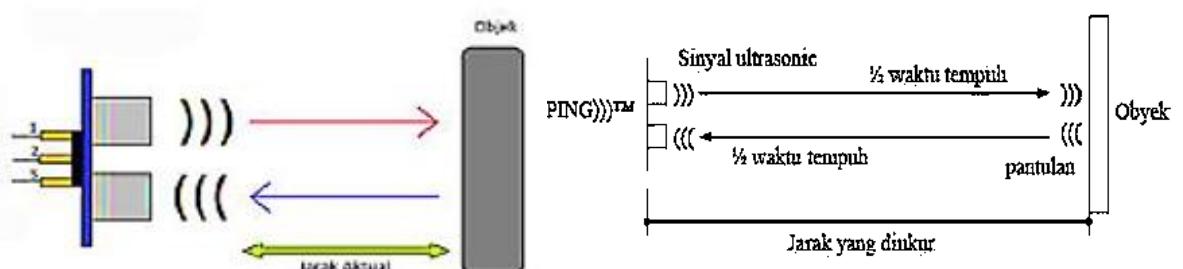
### Fitur sensor HC-SR04 :

<b>Power Supply</b>	+5V DC
<b>Quiescent Current</b>	<2mA
<b>Working Current</b>	15mA
<b>Effectual Angle</b>	<15°
<b>Ranging Distance</b>	2cm – 400 cm/1" - 13ft
<b>Resolution</b>	0.3 cm
<b>Measuring Angle</b>	30 degree
<b>Trigger Input Pulse width</b>	10uS
<b>Dimension</b>	45mm x 20mm x 15mm

## Sistem Kerja Sensor HCSR-04

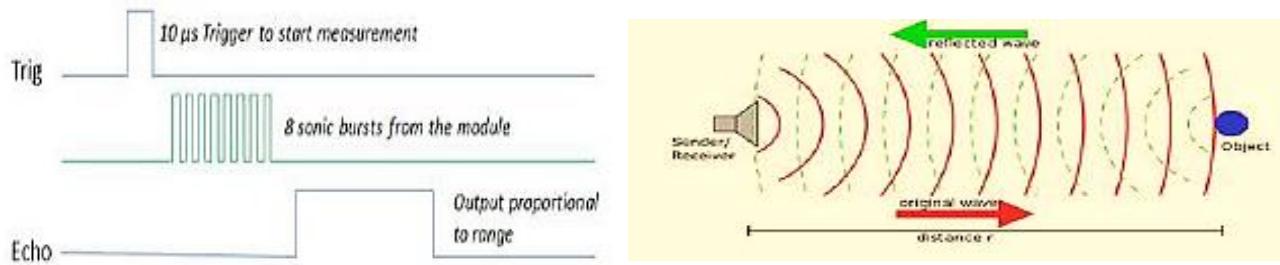
### **Jarak Ukur Sensor PING**

Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz.



Gambar 3. Proses pengukuran jarak

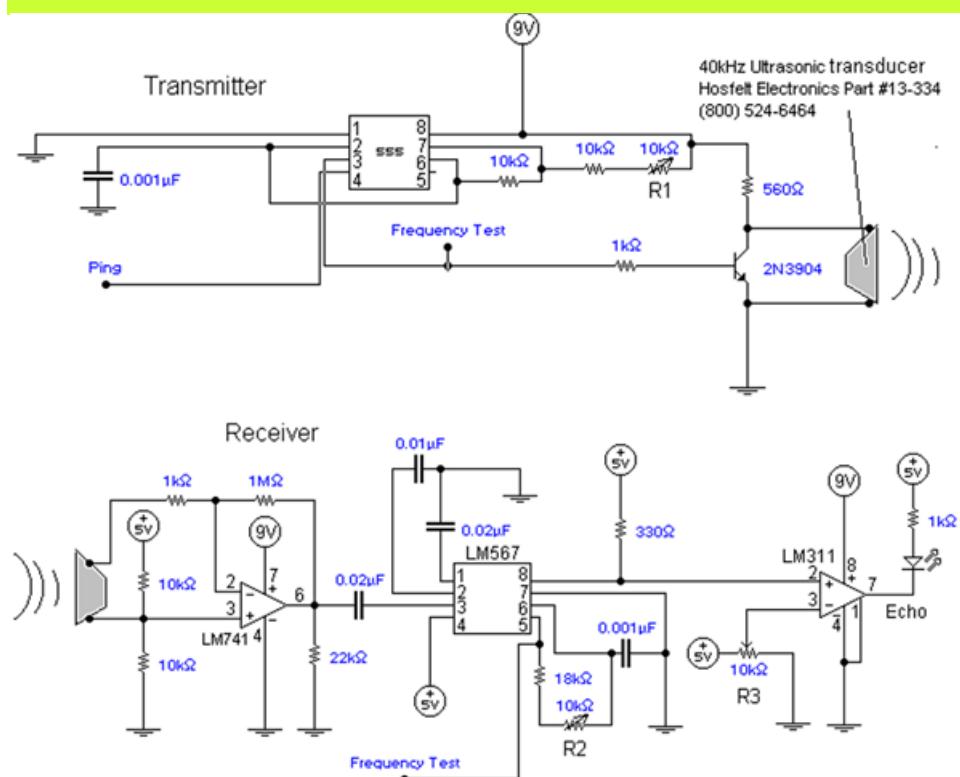
Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.



Gambar 4. Pemancaran gelombang ultrasonik

Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut. Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut.

### Sensor Ultrasonik Pemancar dan Penerima

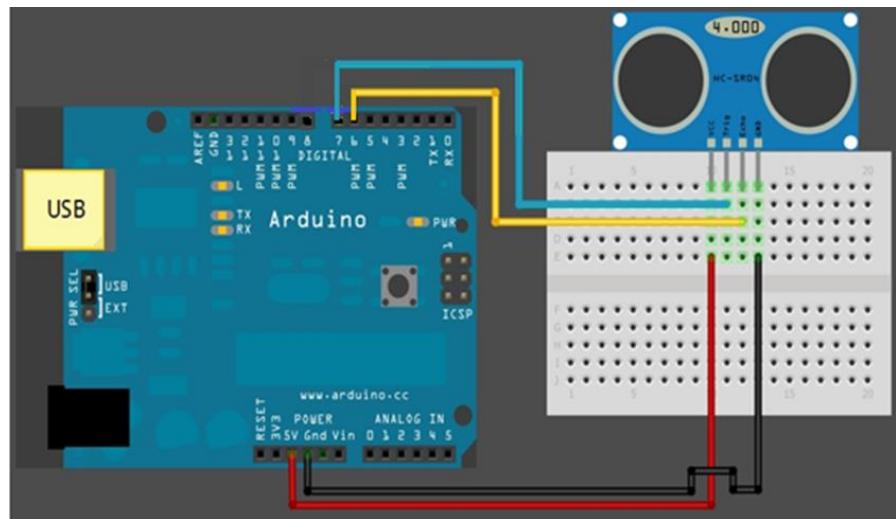


Gambar 5. Skema rangkaian sensor ultrasonic

### C. HARDWARE

- Arduino Uno Board
- 1x Breadboard
- 1x Sensor Ultrasonic
- Kabel jumper

### D. RANGKAIAN



Gambar 6. Rangkaian percobaan

### E. LANGKAH PERCOBAAN

1. Buat rangkaian seperti gambar 6.
2. Aktifkan software IDE arduino.
3. Setting Port COM IDE Arduino dan komputer.
4. Tulis program percobaan ke IDE Arduino
5. Upload dan jalankan program tersebut
6. Amati apakah hasil dari program percobaan sesuai yang diharapkan

### F. PROGRAM PERCOBAAN

#### **Program-1:**

```
#define echoPin 6 // Echo Pin
#define trigPin 7 // Trigger Pin
#define LEDPin 13 // Onboard LED

int maximumRange = 200;
int minimumRange = 0;
long duration, distance;
```

```

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LEDPin, OUTPUT);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration/58.2;

  if (distance >= maximumRange || distance <= minimumRange){
    Serial.println("-1");
    digitalWrite(LEDPin, HIGH);
  }
  else {
    Serial.println(distance);
    digitalWrite(LEDPin, LOW);
  }
  delay(50);
}

```

**Program-2:**

```

const int pwPin = 10;
long pulse, inches, cm;
int measure = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  pinMode(pwPin, INPUT);
  pulse = pulseIn(pwPin, HIGH);
  inches = pulse/147;
  cm = inches*2.54;
  Serial.println(cm); measure = measure + 1;
}

```

**Program-3:**

```

int trigPin = 7; //Trig
int echoPin = 6; //Echo
long duration, cm, meter;
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

```

```
void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    pinMode(echoPin, INPUT);
    duration = pulseIn(echoPin, HIGH);

    // convert jarak → cm dan meter
    cm = (duration/2) / 29.1;
    meter = (duration/2) / 100;
    Serial.print(meter);
    Serial.print("m, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    delay(250);
}
```

## G. LATIHAN

1. Buat aplikasi untuk menentukan jarak benda dengan satuan meter.
2. Buat aplikasi untuk menentukan jarak benda dengan satuan inci (led-1), cm (led-2) dan meter (led-3).

Contoh :

- ketika jarak<2,54 dan jarak >2,54 Led-2 saja yang menyala.
- ketika jarak=2,54 Led-1 saja yang menyala.
- ketika jarak>=100 Led-3 saja yang menyala.

## DAFTAR PUSTAKA

- [1] Julien Bayle, "C Programming for Arduino", Packt Publishing Ltd, Birmingham, May 2013.
- [2] James Floyd K & Harold T , "Arduino Adventure Escape from Gemini Station", Apress, 2013.
- [3] Famosa Studio Arduino Starter Kit Manual – V1.0, Famosa Studio, 2013.
- [4] Martin E, Joshua N, & Jordan H, "Arduino in Action", Maniing Publications.Co, USA, 2013.
- [5] Jack Purdum, "Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers", Apress, 2012.
- [6] John-David Warren, Josh Adams, and Harald Molle, "Arduino Robotics", Springer, New York, 2011.