

Workshop Aplikasi Mikroprosesor & Antarmuka

**PROGRAM STUDI
TEKNIK TELKOMUNIKASI**

Akuwan Saleh, MT

PENILAIAN

- ⇒ Laporan + Tugas + Presentasi = **60%**
 - eval-1(Lap.1-5) = 20%
 - eval-2(Lap.6-10) = 20%
 - eval-3(Lap.11-15 & (PPT+ presentasi)) = 20%
- ⇒ TPS = **40%**
 - eval-4 (TPS = Tugas Proyek Semester)

REFERENSI

- Rui Santos & Sara Santos, “ESP32 Web Server With Arduino IDE.pdf: Step By Step Project Guide”, <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>, juni 2020.
- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013
- Fabian Winkler, “Arduino/Processing Communication Workshop”, Fall, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springer, New York, 2011.
- Casey Reas and Ben Fry, “Getting Started with Processing”, O’Reilly Media, Inc., June 2010.
- Joshua Noble, “Programming Interactivity: A Designer’s Guide to Processing, Arduino, and openFrameworks”, O’Reilly Media, Inc., July 2009.
- Peter Hoddie, Lizzie Prader, “IoT Development for ESP32 and ESP8266 with JavaScript; A Practical Guide to XS and Moddable SDK”, Apress, Menlo Park, CA,USA, 2020.

MATERI

PENDAHULUAN

1. KOMUNIKASI MIKROKONTROLER DENGAN SOFTWARE PROCESSING
2. ANALOG INPUT DAN AUDIO PROCESSING
3. KONTROL MULTI LED MENGGUNAKAN ARDUINO DAN PROCESSING
4. PENCAMPUR WARNA VIRTUAL MENGGUNAKAN ARDUINO DAN PROCESSING
5. MONITORING SUHU DENGAN ARDUINO DAN PROCESSING
6. MONITORING INTENSITAS CAHAYA DENGAN ARDUINO DAN PROCESSING
7. KONTROL MOTOR DC MENGGUNAKAN ARDUINO DAN PROCESSING
8. APLIKASI SENSOR ULTRASONIC MENGGUNAKAN ARDUINO DAN PROCESSING

MATERI

9. KONTROL LAMPU AC 220 V BERBASIS ARDUINO DAN PROCESSING
10. MODUL WiFi ESP32 DENGAN ARDUINO IDE
11. KOMUNIKASI NIRKABEL MENGGUNAKAN MODUL RF 434 MHz DAN PROCESSING
12. ESP32 WEB SERVER UNTUK KONTROL LED DAN MENAMPILKAN GAMBAR
13. ANTARMUKA MODUL GPS DENGAN MIKROKONTROLER DAN PROCESSING
14. ESP32 WEB SERVER UNTUK PENGUKURAN SUHU DAN KELEMBABAN
15. KOMUNIKASI DATA BERBASIS BLUETOOTH DAN HP

8. APLIKASI SENSOR ULTRASONIC MENGUNAKAN ARDUINO DAN PROCESSING

TUJUAN

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

DASAR TEORI

- **Sensor ultrasonic:** kerjanya didasarkan prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi sebuah benda spesifik yang ada dalam frekuensinya.
- **Sensor ultrasonic** adalah sebuah sensor yang memanfaatkan pancaran gelombang ultrasonic.
- **Gelombang ultrasonik** merupakan gelombang akustik yang memiliki frekuensi mulai 20 kHz hingga sekitar 20 MHz.

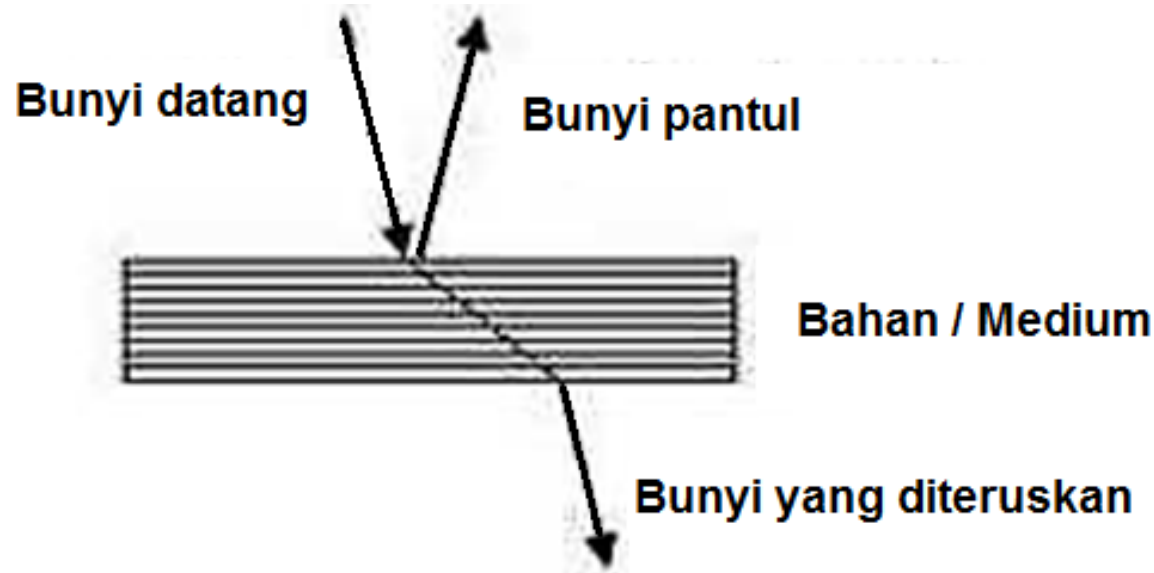
- Selisih waktu ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut.
- Jika gelombang ultrasonik berjalan melalui sebuah medium, Secara matematis besarnya jarak dapat dihitung sebagai berikut:

$$s = v.t/2$$

dimana **s** adalah jarak dalam satuan **meter**, **v** adalah kecepatan gelombang suara yaitu **344 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.



- **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**, **reiceiver**, 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

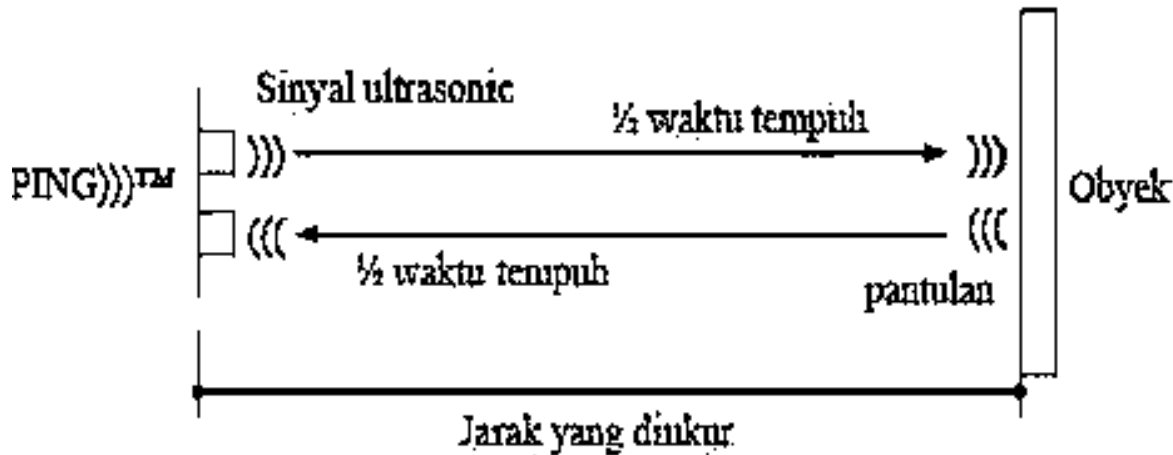
- Receiver terdiri dari transduser ultrasonik menggunakan bahan piezoelektrik, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari transmitter yang dikenakan pada permukaan suatu benda atau gelombang langsung **LOS (Line of Sight)** dari transmitter.

❑ Sensor Ultrasonic PING

- Sensor jarak ultrasonik ping adalah sensor 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, dll.
- Memiliki kemampuan deteksi yang sangat baik dengan akurasi tinggi dan pembacaan stabil.
- Kemampuan mengukur jarak 2cm - 400cm atau 1 inci sampai 4 meter.
- Tidak terpengaruh oleh sinar matahari.

Bentuk sensor ultrasonic



Jarak Ukur Sensor PING

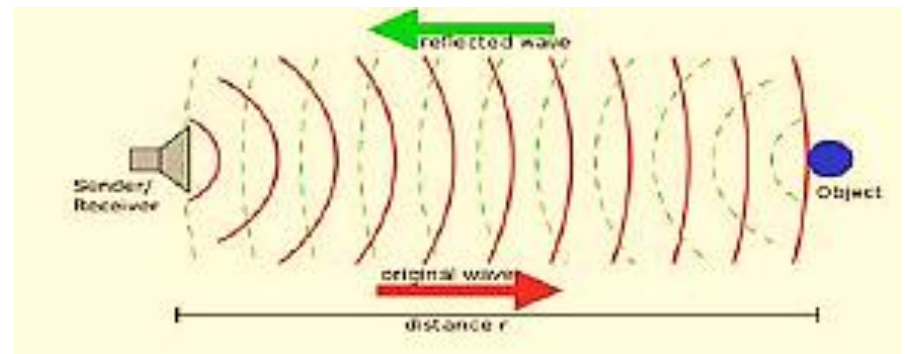
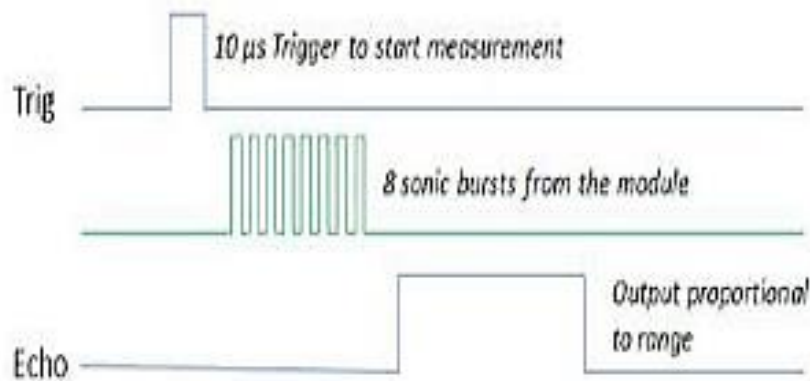
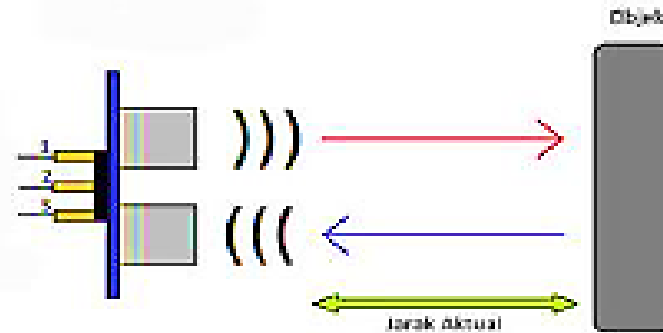
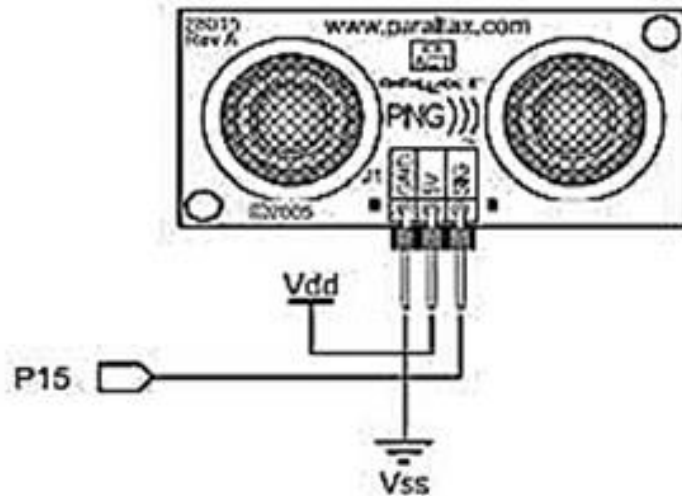
1. **VCC** = 5V Power Supply. Pin sumber tegangan positif sensor.
2. **Trig** = Trigger/Penyulut. Pin ini yang digunakan untuk membangkitkan sinyal ultrasonik.
3. **Echo** = Receive/Indikator. Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik.
4. **GND** = Ground/0V Power Supply. Pin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

Fitur sensor HC-SR04



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

Sistem Kerja Sensor HC-SR04



Sistem Kerja Sensor HC-SR04

- 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.

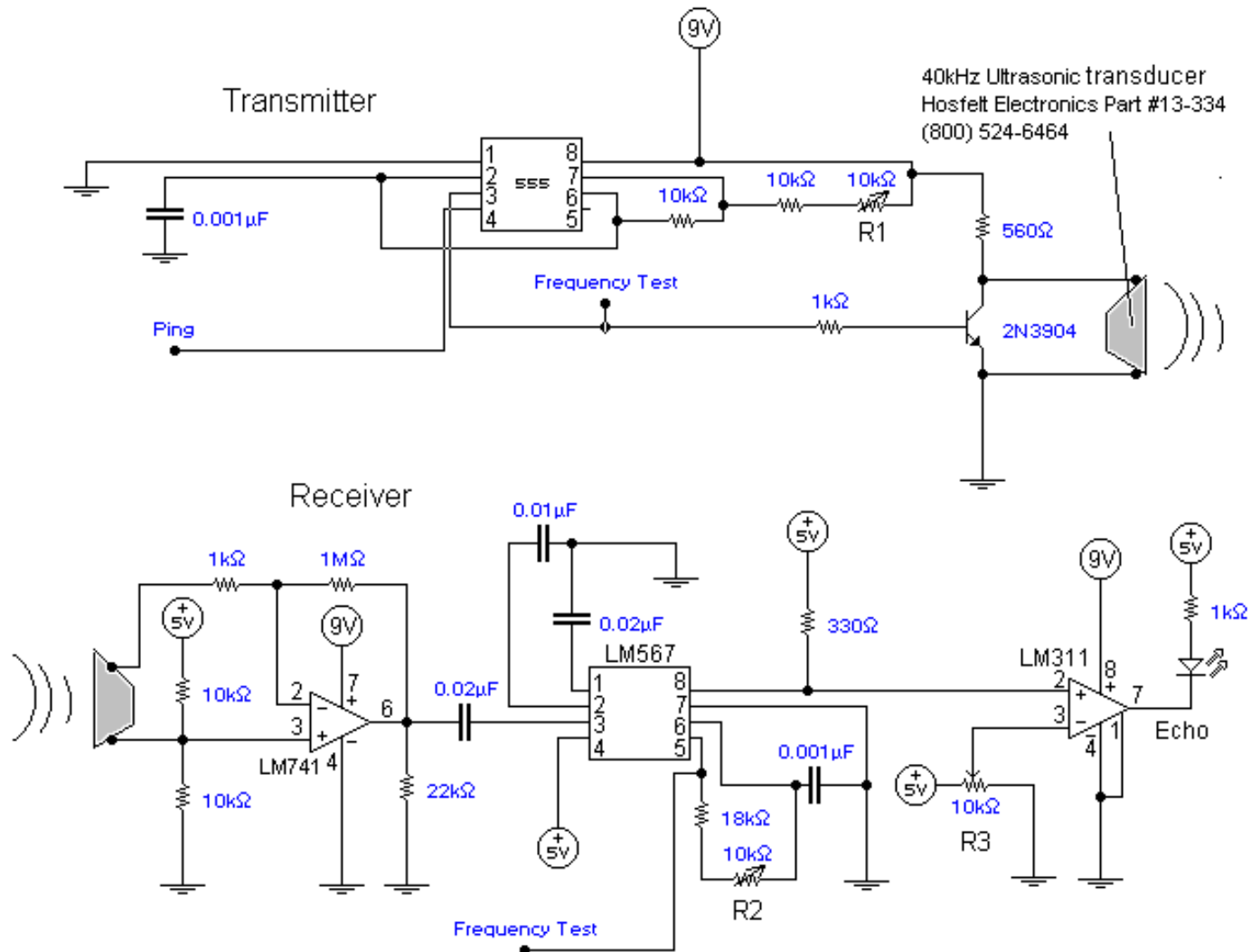
- Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus :

$$S = 340.t/2$$

- Dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang.

Sensor Ultrasonik Pemancar dan Penerima

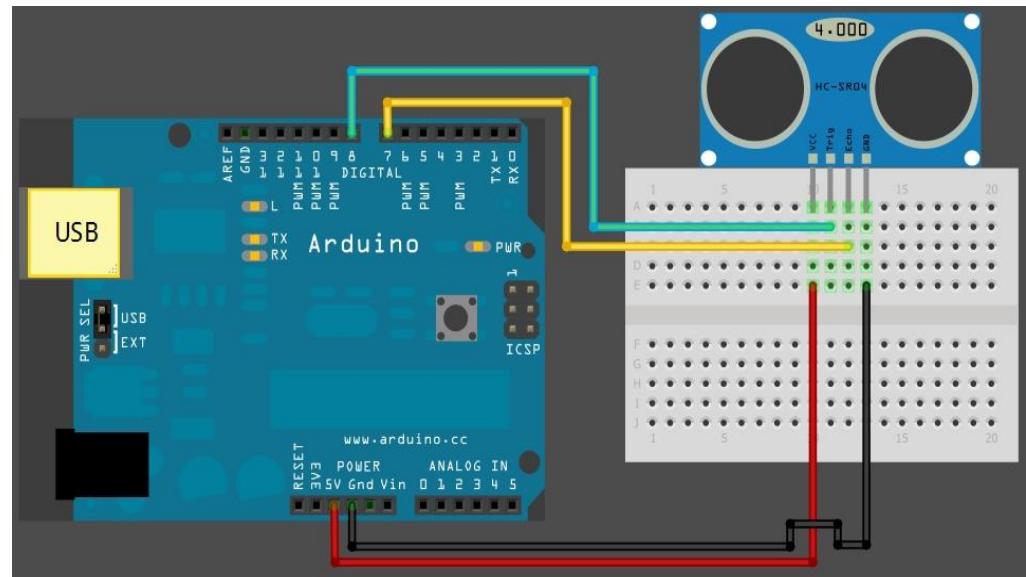
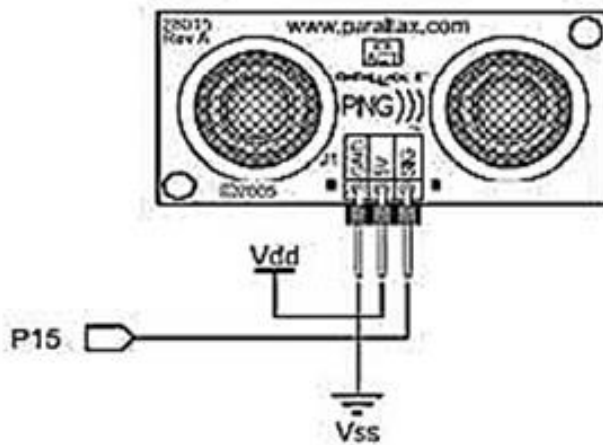
Sonar Ranging System for Minibot



Hardware :

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

Rangkaian:



PROGRAM-1:

Arduino Uno

```
#define echoPin 7 // Echo Pin  
#define trigPin 8 // 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);
```

PROGRAM-1:

Lanjutan Arduino Uno

```
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-1:

Processing

```
import processing.serial.*;

int numOfShapes = 60;
int shapeSpeed = 2;
Square[] mySquares = new Square[numOfShapes];
int shapeSize, distance;
String comPortString;
Serial myPort;
void setup(){
  size(displayWidth,displayHeight);
  smooth();
  shapeSize = (width/numOfShapes);
  for(int i = 0; i<numOfShapes; i++){
    mySquares[i]=new Square(int(shapeSize*i),height-40);
  }
}
```


PROGRAM-1:

Lanjutan Processing

```
myPort = new Serial(this, "COM8", 9600);
myPort.bufferUntil('\n');
}
void draw(){
  background(0);
  delay(50);
  drawSquares();
}
void serialEvent(Serial cPort){
  comPortString = cPort.readStringUntil('\n');
  if(comPortString != null) {
    comPortString=trim(comPortString);
    distance = int(map(Integer.parseInt(comPortString),1,200,1,height));
    if(distance<0){
      distance = 0;
    }
  }
}
```

PROGRAM-1:

Lanjutan Processing

```
void drawSquares(){
int oldY, newY, targetY, redVal, blueVal;
mySquares[0].setY((height-shapeSize)-distance);
for(int i = numOfShapes-1; i>0; i--){
    targetY=mySquares[i-1].getY();
    oldY=mySquares[i].getY();
    if(abs(oldY-targetY)<2){
        newY=targetY;
    }else{
        newY=oldY-((oldY-targetY)/shapeSpeed);
    }
    mySquares[i].setY(newY);
    blueVal = int(map(newY,0,height,0,255));
    redVal = 255-blueVal;
    fill(redVal,0,blueVal);
    rect(mySquares[i].getX(), mySquares[i].getY(),shapeSize,shapeSize);
}
}
```

PROGRAM-1:

Lanjutan Processing

```
boolean sketchFullScreen() {  
  return true;  
}  
class Square{  
  int xPosition, yPosition;  
  Square(int xPos, int yPos){  
    xPosition = xPos;  
    yPosition = yPos;  
  }  
  int getX(){  
    return xPosition;  
  }  
  int getY(){  
    return yPosition;  
  }  
}
```

```
void setY(int yPos){  
  yPosition = yPos;  
}  
}
```

PROGRAM-2:

Arduino Uno

```
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-2:

Processing

```
import processing.serial.*;
import cc.arduino.*;
int lf = 10;
int value;
float numba = map(value, 12, 90, 10, 400);
String myString = null;
Serial myPort;
void setup(){
    printArray(Serial.list());
    background (255);
    size(800, 800);
    myPort = new Serial(this, Serial.list()[0], 9600);
    myString = myPort.readStringUntil(lf);
}
void draw(){
    float numba = map(value, 12, 90, 10, 400);
    stroke(0);
```

PROGRAM-2:

Lanjutan Processing

```
noFill();
drawCircle(width/2, height/2, numba);
branch(width/2, height, 180);
while (myPort.available()>0){
myString = myPort.readStringUntil(lf);
if (myString != null) {
  myString = trim (myString);
  value = int (myString);
  println(numba); }
}
}
void drawCircle(float x, float y, float numba){
ellipse (x, y, numba, numba);
if(numba>2){
  drawCircle(x + numba/1.5, y, numba/2);
  drawCircle(x - numba/1.5, y, numba/2);
}
}
```

PROGRAM-2:

Lanjutan Processing

```
void branch(float x, float y, float numba) {  
  line(x, y, x-numba, y-numba);  
  line(x, y, x+numba, y-numba);  
  if (numba>2){ branch(x-numba, y-numba, numba/2);  
    branch(x+numba, y-numba, numba/2);  
  }  
}
```

Latihan :

1. Buat aplikasi untuk menghitung jarak benda dengan tampilan angka dan animasi persegi panjang.
2. Buat aplikasi untuk menghitung jarak benda dengan tampilan angka dan animasi lingkaran.