

```

#include <Servo.h>
#define Lampu A5
#define reset A0

Servo myservo;

int state = 0;
int LDR= A2;
int LDR1= A3;
int nilaiLDR;
int nilaiLDR1;

int mode1;
int mode2;
int mode3;
int mode;

int status1=0;
//int status1=0;

void setup(){
  Serial.begin(9600);
  myservo.attach(9);
  pinMode(Lampu, OUTPUT);
  pinMode(reset, OUTPUT);

  digitalWrite(Lampu,HIGH);
  digitalWrite(reset,LOW);
}

void loop(){
  baca1();
}

void baca1()
{
  if(Serial.available() > 0){ // Checks
whether data is coming from the serial
port
    state = Serial.read(); // Reads the data
from the serial port
  }

  if (state == 'M'){mode = 1;}
  if (state == '0'){mode = 2;}

  if (mode == 1)
  {
    nilaiLDR= analogRead(LDR);
    nilaiLDR1= analogRead(LDR1);
    float Vout0 = nilaiLDR*0.0048828125;

```

```

float Vout1 = nilaiLDR1*0.0048828125;
int RLDR = (30.0*(5-Vout0))/Vout0;
int RLDR1 = (30.0*(5-Vout1))/Vout1;
delay (500);

if(Serial.available() > 0){ // Checks
whether data is coming from the serial
port
    state = Serial.read(); // Reads the data
from the serial port
}

    if (state == 'A') {
        digitalWrite(Lampu,LOW);
        //Serial.println(RLDR);
        Serial.println("BUKA");
        Serial.println("ON");
        Serial.println(state,BIN);
        delay(1000);
        buka_90_derajat();
        state = 0;
    }

    else if (state == 'B') {
        digitalWrite(Lampu,HIGH);
        //Serial.println(RLDR);
        Serial.println("BUKA");
        Serial.println("OFF");
        Serial.println(state,BIN);
        delay(1000);
        buka_90_derajat();
        state = 0;
    }

    else if (state == 'C') {
        digitalWrite(Lampu,LOW);
        //Serial.println(RLDR);
        Serial.println("TUTUP");
        Serial.println("ON");
        Serial.println(state,BIN);
        delay(1000);
        tutup_90_derajat();
        state = 0;
    }

    else if (state == 'D') {
        digitalWrite(Lampu,HIGH);
        //Serial.println(RLDR);

```

```

Serial.println("TUTUP");
Serial.println("OFF");
Serial.println(state,BIN);
delay(1000);
tutup_90_derajat();
state = 0;
}
if (state == 'O'){mode = 2;}
}
if (mode == 2)
{
nilaiLDR= analogRead(LDR);
nilaiLDR1= analogRead(LDR1);
float Vout0 = nilaiLDR*0.0048828125;
float Vout1 = nilaiLDR1*0.0048828125;
int RLDR = (30.0*(5-Vout0))/Vout0;
int RLDR1 = (30.0*(5-Vout1))/Vout1;
delay (500);

if (RLDR >=20 && RLDR<=500)
{
//Serial.println(RLDR);
}

digitalWrite(Lampu,HIGH);
Serial.println("BUKA");
Serial.println("OFF");
Serial.println(RLDR,BIN);
buka_90_derajat();
while(true)
{
nilaiLDR=
analogRead(LDR);

float Vout0 =
nilaiLDR*0.0048828125;

int RLDR = (30.0*(5-
Vout0))/Vout0;

//Serial.print("LUX= ");
//Serial.println(RLDR);

if(Serial.available() > 0){ // Checks
whether data is coming from the serial
port

state = Serial.read(); // Reads the data
from the serial port

}

if (state == 'E') {

Serial.println(" ");
Serial.println(" ");
Serial.println(state,BIN);
}
}

```

```

digitalWrite(reset,HIGH);
delay(5000);
digitalWrite(reset,HIGH);
delay(3000);
state = 0;
}
delay (500);

if (RLDR >=501 && RLDR
<=1000)
{
//Serial.println(RLDR);
digitalWrite(Lampu,HIGH);
Serial.println("BUKA");
Serial.println("OFF");
Serial.println(RLDR,BIN);
buka_45_derajat();
while(true)
{
nilaiLDR=
analogRead(LDR);

float Vout0 =
nilaiLDR*0.0048828125;

int RLDR = (30.0*(5-
Vout0))/Vout0;

//Serial.print("LUX= ");

//Serial.println(RLDR);

if(Serial.available() > 0){ // Checks
whether data is coming from the serial
port

state = Serial.read(); // Reads the data
from the serial port

}

if (state == 'E') {

Serial.println(" ");
Serial.println(" ");
Serial.println(state,BIN);
delay(5000);
digitalWrite(reset,HIGH);
delay(3000);
state = 0;
}

delay (500);

if (RLDR <=10)
{

```

```

//Serial.println(RLDR);
Serial.println("TUTUP");
Serial.println("ON");
Serial.println(RLDR,BIN);
buka_45_derajat();
while(true)
{
    digitalWrite(Lampu,LOW);

    nilaiLDR=
analogRead(LDR);

    float Vout0 =
nilaiLDR*0.0048828125;

    int RLDR = (30.0*(5-
Vout0))/Vout0;

    //Serial.print("LUX= ");

    //Serial.println(RLDR);

    if(Serial.available() > 0){ // Checks
whether data is coming from the serial
port

    state = Serial.read(); // Reads the data
from the serial port

    }

    if (state == 'E') {

Serial.println(" ");
Serial.println(" ");
Serial.println(state,BIN);
delay(5000);
digitalWrite(reset,HIGH);
delay(3000);
state = 0;
}

    delay (500);

if (RLDR >=20 && RLDR<=500)
{
//Serial.println(RLDR);
digitalWrite(Lampu,HIGH);
Serial.println("BUKA");
Serial.println("OFF");
Serial.println(RLDR,BIN);
tutup_90_derajat();
while(true)
{
    nilaiLDR=
analogRead(LDR);

```

```

float Vout0 =
nilaiLDR*0.0048828125;

int RLDR = (30.0*(5-
Vout0))/Vout0;

//Serial.print("LUX= ");

//Serial.println(RLDR);

if(Serial.available() > 0){ // Checks
whether data is coming from the serial
port

state = Serial.read(); // Reads the data
from the serial port

}

if (state == 'E') {

Serial.println(" ");

Serial.println(" ");

Serial.println(state,BIN);

delay(5000);

digitalWrite(reset,HIGH);

delay(3000);

state = 0;

}

delay (500);

if (RLDR >=501 && RLDR
<=1000)

```

```

{

//Serial.println(RLDR);

digitalWrite(Lampu,HIGH);

Serial.println("BUKA");

Serial.println("OFF");

Serial.println(RLDR,BIN);

tutup_45_derajat();

while(true)

{

nilaiLDR=
analogRead(LDR);

float Vout0 =
nilaiLDR*0.0048828125;

int RLDR = (30.0*(5-
Vout0))/Vout0;

//Serial.print("LUX= ");

//Serial.println(RLDR);

if(Serial.available() > 0){ // Checks
whether data is coming from the serial
port

state = Serial.read(); // Reads the data
from the serial port

}

if (state == 'E') {

```

```

Serial.println(" ");
Serial.println(" ");
Serial.println(state,BIN);
delay(5000);
digitalWrite(reset,HIGH);
delay(3000);
state = 0;
}

    delay (500);

if (RLDR <=10)
{
    //Serial.println(RLDR);
    digitalWrite(Lampu,LOW);
    Serial.println("TUTUP");
    Serial.println("ON");
    Serial.println(RLDR,BIN);
    tutup_45_derajat();

    while(true)
    {
        nilaiLDR=
analogRead(LDR);

float Vout0 =
nilaiLDR*0.0048828125;

int RLDR = (30.0*(5-
Vout0))/Vout0;

//Serial.print("LUX= ");
//Serial.println(RLDR);

if(Serial.available() > 0){ // Checks
whether data is comming from the serial
port

state = Serial.read(); // Reads the data
from the serial port

}

if (state == 'E') {

Serial.println(" ");
Serial.println(" ");
Serial.println(state,BIN);
delay(5000);
digitalWrite(reset,HIGH);
delay(3000);

state = 0;

}

delay (500);

    baca1();
}
}

```

```

    }           myservo.write(0);
  }           delay(1000);
}           myservo.detach();
}           delay(1000);
}
}           // myservo.write(0);
}           // Serial.println("servo off 45");
}
}
}           void buka_45_derajat()
}           {
}           myservo.attach(9);
}
}
}           //buka 45 derajat
void tutup_45_derajat()           myservo.write(500);
{           delay(5000);
myservo.attach(9);           myservo.detach();
           delay(1000);
//tutup 45 derajat           myservo.write(500);
myservo.write(0);           delay(1000);
delay(5000);           myservo.detach();
myservo.detach();           delay(1000);
delay(1000);           //myservo.write(0);

```



```

// Serial.println("servo buka 45");
}

}

void buka_90_derajat()
{
myservo.attach(9);
//buka 90 derajat
myservo.write(500);
delay(18000);
myservo.detach();
delay(1000);
myservo.write(500);
delay(1000);
myservo.detach();
delay(1000);
// myservo.write(0);
// Serial.println("servo buka 90");
}

void tutup_90_derajat()
{
myservo.attach(9);
//tutup 90 derajat
myservo.write(0);
delay(18000);
myservo.detach();
delay(1000);
myservo.write(0);
delay(1000);
myservo.detach();
delay(1000);
// myservo.write(0);
// Serial.println("servo off 90");
}

```

## Program Aplikasi Android dengan MIT App Inventor

The image displays several MIT App Inventor code blocks for a Bluetooth application:

- when ListPicker1.Picked:** A 'do' block containing 'set ListPicker1.Elements to BluetoothClient1.AddressesAndNames'.
- when ListPicker1.AfterPicking:** An 'if' block. The 'if' branch calls 'BluetoothClient1.Connect' with 'address' set to 'ListPicker1.Selection', followed by 'set ListPicker1.Elements to BluetoothClient1.AddressesAndNames'. The 'then' branch sets 'Hasil\_Bluetooth.Text' to 'Connected' and 'Hasil\_Bluetooth.TextColor' to white. The 'else' branch sets 'Hasil\_Bluetooth.Text' to 'Not Connected' and 'Hasil\_Bluetooth.TextColor' to red.
- when Screen1.BackPressed:** A 'do' block containing 'close application'.
- when Clock1.Timer:** An 'if' block. The 'if' branch is 'BluetoothClient1.IsConnected' and 'true'. It contains 'call BluetoothClient1.BytesAvailableToReceive' and 'then' 'set Hasil.Text to call BluetoothClient1.ReceiveText' with 'numberOfBytes' set to 'call BluetoothClient1.BytesAvailableToReceive'.
- when Minus1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '1'.
- when Lampu\_Merah1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '2'.
- when Lampu\_Hijau1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '3'.
- when Lampu\_Kuning1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '4'.
- when Lampu\_Merah1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '5'.
- when Lampu\_Hijau1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '6'.
- when Lampu\_Kuning1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '7'.
- when Lampu\_Merah1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '8'.
- when Lampu\_Hijau1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '9'.
- when Lampu\_Kuning1.Click:** A 'do' block containing 'call BluetoothClient1.SendText' with 'text' set to '0'.

## Tampilan Aplikasi Sistem Kendali Tirai Vertical Blind dan Lampu Ruangan



### Tombol Bluetooth Ditekan



### Bluetooth dan Alat Tersambung



### Tombol Manual Ditekan

### Tombol Tirai Tertutup Lampu Hidup Diteka

**BENTUK FISIK ALAT**



**TIRAI TERBUKA LAMPU HIDUP**



**TIRAI TERTUTUP LAMPU MATI**



**TIRAI TERBUKA LAMPU MATI**

ALAT PENGENDALI



**TIRAI TERTUTUP LAMPU HIDUP**