

PROGRAM KURSI RODA ELEKTRIK BERBASIS ARDUINO MEGA 2560

```
// URTouch_ButtonTest
// This program is a quick demo of how create and use buttons.
// This program requires the UTFT library.
// It is assumed that the display module is connected to an
// appropriate shield or that you know how to change the pin
// numbers in the setup.
#include <UTFT.h>
#include <URTouch.h>
// Define the orientation of the touch screen. Further
// information can be found in the instructions.
#define TOUCH_ORIENTATION PORTRAIT
// Initialize display
// -----
// Set the pins to the correct ones for your development board
// -----
// Standard Arduino Uno/2009 Shield      : <display model>,19,18,17,16
// Standard Arduino Mega/Due shield      : <display model>,38,39,40,41
// CTE TFT LCD/SD Shield for Arduino Due  : <display model>,25,26,27,28
// Teensy 3.x TFT Test Board             : <display model>,23,22, 3, 4
// ElecHouse TFT LCD/SD Shield for Arduino Due : <display
model>,22,23,31,33
//
// Remember to change the model parameter to suit your display module!
UTFT myGLCD(ITDB32WC,38,39,40,41);
// Initialize touchscreen
// -----
// Set the pins to the correct ones for your development board
// -----
// Standard Arduino Uno/2009 Shield      : 15,10,14, 9, 8
// Standard Arduino Mega/Due shield      : 6, 5, 4, 3, 2
// CTE TFT LCD/SD Shield for Arduino Due  : 6, 5, 4, 3, 2
// Teensy 3.x TFT Test Board             : 26,31,27,28,29
// ElecHouse TFT LCD/SD Shield for Arduino Due : 25,26,27,29,30
//
URTouch myTouch( 6, 5, 4, 3, 2);
// Declare which fonts we will be using
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];
uint32_t cx, cy;
uint32_t rx[8], ry[8];
uint32_t clx, crx, cty, cby;
float px, py;
int dispx, dispy, text_y_center;
```

```

uint32_t calx, caly, calz;
char buf[13];
bool setnewcode_flag = 0;
bool setoldcode_flag = 0;
bool repeat_flag = 1;
bool lock_flag = 0;
int x, y, z;
String stCode = "12345";
String stTemp = "";
char stCurrent[6] = ""; //array 6 dimensi untuk menyimpan inputan angka bukan
password
char asterisc[6] = ""; // hidden password dengan bentuk bintang - bintang
char asterisc_partial[6] = "";
String stLast = "";
String msgCurrent = ""; //pesan blink tulisan ke lcd
int stCurrentLen=0; // menghapus memori sementara
int thisByte = 0; // data serial hexa desimal
int data_sent = 0; // data yang di kirim
int buttonState = 0; //kondisi awal button adalah 0
int SW_pin = 8; // brake
int sensorPin = A0; // select the input pin for the photodiode
int sensorValue = 0; // variable to store the value coming from the sensor
const int X_pin = 9; // analog pin connected to X output
const int Y_pin = 10; // analog pin connected to Y output
int kiri1 = 10; // kecepatan motor
int kanan1 = 9;
int LR = 11; // rotasi motor
int RR = 12;
int gigimundur = 18;
int power = 13;
int nilai;
int gigimundurState = 0;
int powerState = 0;
int count=0;
int val; // simpan nilai kesalahan
int stLastserial;
/*****
** Custom functions **
*****/
void cleardisplay () //kotak hapus angka inputan
{
  myGLCD.setColor(230, 230, 230);
  //myGLCD.fillRect(0, 0, 239, 50); // clear display
  myGLCD.fillRect(30, 230, 285, 190); // clear display untuk kotak besar //fix
}
void clearsmalldisplay () //menghapus angka pwd dan tanda bintang
{
  myGLCD.setColor(255, 255, 255);

```

```

//myGLCD.fillCircle(51, 27, 16);
myGLCD.fillCircle(120, 210, 16); //set lingkaran kecil tempat pwd kiri // FIX
//myGLCD.fillCircle(188, 27, 16);
myGLCD.fillCircle(200, 210, 16); //set lingkaran kecil tempat pwd kanan //FIX
//myGLCD.fillRect(51, 11, 188, 43); // clear small display
myGLCD.fillRect(120, 225, 200, 195); // clear small display //FIX
}
void drawButtons()
{
// Draw the upper row of buttons
for (x=0; x<5; x++)
{
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (0+(x*100), 0, 60+(x*100), 60);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (0+(x*100), 0, 60+(x*100), 60);
myGLCD.printNumI(x+1, 27+(x*100), 27);
}
// Draw the center row of buttons
for (x=0; x<5; x++)
{
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (0+(x*100), 80, 60+(x*100), 140);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (0+(x*100), 80, 60+(x*100), 140);
if (x<4)
myGLCD.printNumI(x+6, 27+(x*100), 107);
}
myGLCD.print("0", 427, 107);
// Draw the lower row of buttons
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (0, 160, 230, 240);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (0, 160, 230, 240);
myGLCD.print("CLEAR", 115, 190);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (240, 160, 460, 240);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (240, 160, 460, 240);
myGLCD.print("ENTER", 365, 190);
myGLCD.setBackgroundColor (0, 0, 0);
}
void blinkmsg (String msg, String color) //blok membuat tulisan dan warna
{
myGLCD.setBackgroundColor (290, 290, 290);
if (color == "RED")
{
myGLCD.setColor(255, 0, 0);

```

```

}
if (color == "WHITE")
{
    myGLCD.setColor(255, 255, 255);
}
if (color == "GREEN")
{
    myGLCD.setColor(0, 255, 0);
}
if (color == "BLACK")
{
    myGLCD.setColor(0, 0, 0);
}
myGLCD.print(msg, CENTER, 290);          // blink msg
delay(500);
myGLCD.print("          ", CENTER, 290);
delay(500);
myGLCD.print(msg, CENTER, 290);
delay(500);
myGLCD.print("          ", CENTER, 290);
delay(500);
}
void printmsg (String msg, String color) //blok bagian menampilkan
{
    if (color == "WHITE")
    {
        myGLCD.setBackColor (0, 0, 0);
        myGLCD.setColor(255, 255, 255);
    }
    if (color == "GREEN")
    {
        myGLCD.setBackColor (290, 290, 290);
        myGLCD.setColor(0, 255, 0);
    }
    if (color == "RED")
    {
        myGLCD.setBackColor (290, 290, 290);
        myGLCD.setColor(255, 0, 0);
    }
    if (color == "BLACK")
    {
        myGLCD.setBackColor (290, 290, 290);
        myGLCD.setColor(0, 0, 0);
    }
    myGLCD.print(msg, CENTER, 290);
}
void updateStr(int val)
{

```

```

if (stCurrentLen<5) //sesuaikan dengan jumlah pwd
{
  if (stCurrentLen==0)
  {
    cleardisplay();
    clearsmalldisplay();
  }
  else
  clearsmalldisplay();
  stCurrent[stCurrentLen]=val;
  stCurrent[stCurrentLen+1]='\0';
  asterisc[stCurrentLen]='*';
  asterisc[stCurrentLen+1]='\0';
  if (stCurrentLen>0)
  {
    asterisc_partial[stCurrentLen]=val;
    asterisc_partial[stCurrentLen+1]='\0';
    asterisc_partial[stCurrentLen-1]='*';
  }
  else
  {
    asterisc_partial[stCurrentLen]=val;
    asterisc_partial[stCurrentLen+1]='\0';
  }
  stCurrentLen++;
  myGLCD.setColor(0, 0, 0);
  myGLCD.setBackColor (255, 255, 255);
  //myGLCD.print("_____", CENTER, 20);
  myGLCD.print("_____", CENTER, 290); //FIX
  //myGLCD.print(asterisc_partial, 56, 20);
  myGLCD.print(asterisc_partial, 120, 290); //set muncul angka pwd //FIX
}
}
// Draw a red frame while a button is touched
void waitForIt(int x1, int y1, int x2, int y2)
{
  myGLCD.setColor(255, 0, 0);
  myGLCD.drawRoundRect (x1, y1, x2, y2);
  while (myTouch.dataAvailable())
    myTouch.read();
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (x1, y1, x2, y2);
}
/*****
** Required functions **
*****/
void setup()
{

```

```

pinMode(SW_pin, INPUT);
digitalWrite(SW_pin, HIGH);
pinMode(kiri1, OUTPUT);
pinMode(kanan1, OUTPUT);
pinMode(LR, OUTPUT);
pinMode(RR, OUTPUT);
pinMode(gigimundur, OUTPUT);
pinMode(power, OUTPUT);
Serial.begin(115200);
{
  randomSeed(analogRead(0));
// Setup the LCD
  myGLCD.InitLCD();
  myGLCD.clrScr();
  myGLCD.setFont(BigFont);
  startup();
  delay(15000);
}
{
  myGLCD.InitLCD();
  myGLCD.clrScr();
  myGLCD.setFont(BigFont);
  for (int i=0; i<=100; i++){
    myGLCD.setColor(15,55+(2*i),0);
    myGLCD.print("Loading...", 100, 160, 0);
    myGLCD.setFont(SevenSegNumFont);
    myGLCD.printNumI(i,270,127,0);
    myGLCD.setFont(BigFont);
    if(i<=99){myGLCD.print("%", 345, 160, 0);}
    if(i>99){myGLCD.print("%", 375, 160, 0);}
  }
  myGLCD.setColor(0, 0, 255);
  myGLCD.print("KURSI RODA ELEKTRIK", 0, 06, 0);
  myGLCD.setColor(0, 255, 0);
  myGLCD.print("by M.Adrian Saputra", 479, 0, 90);
  myGLCD.setColor(2550, 0, 0);
  myGLCD.print("POLITEKNIK NEGERI SRIWIJAYA", 50, 300, 0);
  myGLCD.setColor(255, 0, 255);
  myGLCD.print("M.MUKHLIS D.P", 0, 319, 270);
  /*
  myGLCD.setFont(SevenSegNumFont);
  myGLCD.setColor(0, 255, 0);
  myGLCD.print("45", 65, 75, 45);
  myGLCD.print("90", 350, 50, 90);
  myGLCD.print("180", 420, 250, 180);
  */
  delay(2000);
}

```

```

// Initial setup
myGLCD.InitLCD();
myGLCD.clrScr();
myTouch.InitTouch();
myTouch.setPrecision(PREC_MEDIUM);
myTouch.InitTouch(TOUCH_ORIENTATION);
dispx=myGLCD.getDisplayXSize();
dispy=myGLCD.getDisplayYSize();
text_y_center=(dispy/2)-6;
myGLCD.setFont(BigFont);
myGLCD.setBackColor(0, 0, 255);
drawButtons();
}
void startup()
{
  myGLCD.setColor(255, 0, 0);
  myGLCD.fillRect(0, 0, dispx-1, 13);
  myGLCD.setColor(255, 255, 255);
  myGLCD.setBackColor(255, 0, 0);
  myGLCD.drawLine(0, 14, dispx-1, 14);
  myGLCD.print("KURSI RODA ELEKTRIK", CENTER, 1);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.print("INSTRUKSI", CENTER, 30);
  myGLCD.print("Masukkan Password", LEFT, 76);
  myGLCD.print("Kemudian Tekan Tombol Enter", LEFT, 94);
  myGLCD.print("Password Benar Joystick Fungsi", LEFT, 112);
  myGLCD.print("Tekan Password Lagi Untuk", LEFT, 130);
  myGLCD.print("Mematikan Joystick", LEFT, 148);
  myGLCD.print("JANGAN LUPA PASSWORD", CENTER, 170);
  myGLCD.print("KEAMANAN NO.1 !!!", CENTER, 187);
  myGLCD.print("INGAT JANGAN SAMPAI SALAH", CENTER, 226);
}
void loop()
{
  while (true)
  {
    powerState = digitalRead(power);
    if (analogRead(X_pin)<600 && analogRead(X_pin)>400 &&
    analogRead(Y_pin)>400 && analogRead(Y_pin)<600 && (powerState ==
    HIGH))
    {
      Serial.println("\n\nnetral");
      digitalWrite(LR, HIGH);
      digitalWrite(RR, HIGH);
      analogWrite(kanan1, 0);
      analogWrite(kiri1, 0);
      delay(50);}
  }
}

```

```

    if (analogRead(X_pin)<401 && analogRead(Y_pin)>400 &&
analogRead(Y_pin)<660 && (powerState == HIGH)){
    Serial.println("\n\nmaju");
    analogWrite(kanan1, 120);
    analogWrite(kiri1, 120);
    Serial.println(nilai);
    digitalWrite(LR, HIGH);
    digitalWrite(RR, HIGH);
    }
    if (analogRead(Y_pin)>600 && analogRead(X_pin)>400 &&
analogRead(X_pin)<660 && (powerState == HIGH)){
    Serial.println("\n\nkiri");
    analogWrite(kanan1, 120);
    analogWrite(kiri1, 120);
    digitalWrite(LR, HIGH);
    digitalWrite(RR, LOW);

    }
    if (analogRead(Y_pin)<401 && analogRead(X_pin)>400 &&
analogRead(X_pin)<660 && (powerState == HIGH)){
    Serial.println("\n\nkanan");
    analogWrite(kanan1, 120);
    analogWrite(kiri1,120);
    digitalWrite(RR, HIGH);
    digitalWrite(LR, LOW);
    }
    if (analogRead(X_pin)>600 && analogRead(Y_pin)>400 &&
analogRead(Y_pin)<660 && (powerState == HIGH)){
    Serial.println("\n\nmundur");
    analogWrite(kanan1, 120);
    analogWrite(kiri1, 120);
    Serial.println(nilai);
    digitalWrite(LR, LOW);
    digitalWrite(RR, LOW);
    }
    if (digitalRead(SW_pin)<1){
    Serial.println("\n\nnetral");
    digitalWrite(gigimundur, LOW);
    analogWrite(kanan1, 0);
    analogWrite(kiri1, 0);
    digitalWrite(LR, HIGH);
    digitalWrite(RR, HIGH);
    }
    if (myTouch.dataAvailable())
    {
    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

```

```

if ((y>=0) && (y<=7)) // Upper row
{
if ((x>=0) && (x<=30)) // Button: 1
{
waitForIt(0, 0, 60, 60);
updateStr('1');
Serial.print("data=1"); // print as an ASCII-encoded decimal - same as "DEC"
Serial.print("\t"); // prints a tab
Serial.print("DEC=");
Serial.print(49, DEC); // print as an ASCII-encoded decimal
Serial.print("\t"); // prints a tab
Serial.print("HEX=");
Serial.print(49, HEX); // print as an ASCII-encoded hexadecimal
Serial.print("\t"); // prints a tab
Serial.print("OCT=");
Serial.print(49, OCT); // print as an ASCII-encoded octal
Serial.print("\t"); // prints a tab
Serial.print("BIN=");
Serial.println(49, BIN); // print as an ASCII-encoded binary
}
if ((x>=46) && (x<=75)) // Button: 2
{
waitForIt(100, 0, 160, 60);
updateStr('2');
Serial.print("data=2"); // print as an ASCII-encoded decimal - same as "DEC"
Serial.print("\t"); // prints a tab
Serial.print("DEC=");
Serial.print(50, DEC); // print as an ASCII-encoded decimal
Serial.print("\t"); // prints a tab
Serial.print("HEX=");
Serial.print(50, HEX); // print as an ASCII-encoded hexadecimal
Serial.print("\t"); // prints a tab
Serial.print("OCT=");
Serial.print(50, OCT); // print as an ASCII-encoded octal
Serial.print("\t"); // prints a tab
Serial.print("BIN=");
Serial.println(50, BIN); // print as an ASCII-encoded binary
}
if ((x>=76) && (x<=110)) // Button: 3
{
waitForIt(200, 0, 260, 60);
updateStr('3');
Serial.print("data=3"); // print as an ASCII-encoded decimal - same as "DEC"
Serial.print("\t"); // prints a tab
Serial.print("DEC=");
Serial.print(51, DEC); // print as an ASCII-encoded decimal
Serial.print("\t"); // prints a tab
Serial.print("HEX=");

```

```

Serial.print(51, HEX); // print as an ASCII-encoded hexadecimal
Serial.print("\t"); // prints a tab
Serial.print("OCT=");
Serial.print(51, OCT); // print as an ASCII-encoded octal
Serial.print("\t"); // prints a tab
Serial.print("BIN=");
Serial.println(51, BIN); // print as an ASCII-encoded binary
}
if ((x>=111) && (x<=149)) // Button: 4
{
  waitForIt(300, 0, 360, 60);
  updateStr('4');
  Serial.print("data=4"); // print as an ASCII-encoded decimal - same as "DEC"
  Serial.print("\t"); // prints a tab
  Serial.print("DEC=");
  Serial.print(52, DEC); // print as an ASCII-encoded decimal
  Serial.print("\t"); // prints a tab
  Serial.print("HEX=");
  Serial.print(52, HEX); // print as an ASCII-encoded hexadecimal
  Serial.print("\t"); // prints a tab
  Serial.print("OCT=");
  Serial.print(52, OCT); // print as an ASCII-encoded octal
  Serial.print("\t"); // prints a tab
  Serial.print("BIN=");
  Serial.println(52, BIN); // print as an ASCII-encoded binary
}
if ((x>=150) && (x<=180)) // Button: 5
{
  waitForIt(400, 0, 460, 60);
  updateStr('5');
  Serial.print("data=5"); // print as an ASCII-encoded decimal - same as "DEC"
  Serial.print("\t"); // prints a tab
  Serial.print("DEC=");
  Serial.print(53, DEC); // print as an ASCII-encoded decimal
  Serial.print("\t"); // prints a tab
  Serial.print("HEX=");
  Serial.print(53, HEX); // print as an ASCII-encoded hexadecimal
  Serial.print("\t"); // prints a tab
  Serial.print("OCT=");
  Serial.print(53, OCT); // print as an ASCII-encoded octal
  Serial.print("\t"); // prints a tab
  Serial.print("BIN=");
  Serial.println(53, BIN); // print as an ASCII-encoded binary
}
}
if ((y>=8) && (y<=15)) // Center row
{
  if ((x>=0) && (x<30)) // Button: 6

```

```

{
  waitForIt(0, 80, 60, 140);
  updateStr('6');
  Serial.print("data=6"); // print as an ASCII-encoded decimal - same as "DEC"
  Serial.print("\t"); // prints a tab
  Serial.print("DEC=");
  Serial.print(54, DEC); // print as an ASCII-encoded decimal
  Serial.print("\t"); // prints a tab
  Serial.print("HEX=");
  Serial.print(54, HEX); // print as an ASCII-encoded hexadecimal
  Serial.print("\t"); // prints a tab
  Serial.print("OCT=");
  Serial.print(54, OCT); // print as an ASCII-encoded octal
  Serial.print("\t"); // prints a tab
  Serial.print("BIN=");
  Serial.println(54, BIN); // print as an ASCII-encoded binary
}
if ((x>=46) && (x<=75)) // Button: 7
{
  waitForIt(100, 80, 160, 140);
  updateStr('7');
  Serial.print("data=7"); // print as an ASCII-encoded decimal - same as "DEC"
  Serial.print("\t"); // prints a tab
  Serial.print("DEC=");
  Serial.print(55, DEC); // print as an ASCII-encoded decimal
  Serial.print("\t"); // prints a tab
  Serial.print("HEX=");
  Serial.print(55, HEX); // print as an ASCII-encoded hexadecimal
  Serial.print("\t"); // prints a tab
  Serial.print("OCT=");
  Serial.print(55, OCT); // print as an ASCII-encoded octal
  Serial.print("\t"); // prints a tab
  Serial.print("BIN=");
  Serial.println(55, BIN); // print as an ASCII-encoded binary
}
if ((x>=76) && (x<=110)) // Button: 8
{
  waitForIt(200, 80, 260, 140);
  updateStr('8');
  Serial.print("data=8"); // print as an ASCII-encoded decimal - same as "DEC"
  Serial.print("\t"); // prints a tab
  Serial.print("DEC=");
  Serial.print(56, DEC); // print as an ASCII-encoded decimal
  Serial.print("\t"); // prints a tab
  Serial.print("HEX=");
  Serial.print(56, HEX); // print as an ASCII-encoded hexadecimal
  Serial.print("\t"); // prints a tab
  Serial.print("OCT=");

```

```

Serial.print(56, OCT); // print as an ASCII-encoded octal
Serial.print("\t"); // prints a tab
Serial.print("BIN=");
Serial.println(56, BIN); // print as an ASCII-encoded binary
}
if ((x>=111) && (x<=149)) // Button: 9
{
  waitForIt(300, 80, 360, 140);
  updateStr('9');
  Serial.print("data=9"); // print as an ASCII-encoded decimal - same as "DEC"
  Serial.print("\t"); // prints a tab
  Serial.print("DEC=");
  Serial.print(57, DEC); // print as an ASCII-encoded decimal
  Serial.print("\t"); // prints a tab
  Serial.print("HEX=");
  Serial.print(57, HEX); // print as an ASCII-encoded hexadecimal
  Serial.print("\t"); // prints a tab
  Serial.print("OCT=");
  Serial.print(57, OCT); // print as an ASCII-encoded octal
  Serial.print("\t"); // prints a tab
  Serial.print("BIN=");
  Serial.println(57, BIN); // print as an ASCII-encoded binary
}
if ((x>=110) && (x<=180)) // Button: 0
{
  waitForIt(400, 80, 460, 140);
  updateStr('0');
  Serial.print("data=0"); // print as an ASCII-encoded decimal - same as "DEC"
  Serial.print("\t"); // prints a tab
  Serial.print("DEC=");
  Serial.print(48, DEC); // print as an ASCII-encoded decimal
  Serial.print("\t"); // prints a tab
  Serial.print("HEX=");
  Serial.print(48, HEX); // print as an ASCII-encoded hexadecimal
  Serial.print("\t"); // prints a tab
  Serial.print("OCT=");
  Serial.print(48, OCT); // print as an ASCII-encoded octal
  Serial.print("\t"); // prints a tab
  Serial.print("BIN=");
  Serial.println(48, BIN); // print as an ASCII-encoded binary
  myGLCD.setColor(0, 0, 0);
  myGLCD.setBackColor(255, 255, 255);
  //myGLCD.print(asterisc, 56, 20);
  myGLCD.print(asterisc, 120, 290); //FIX piksel dan tampilan bintang kejora
}
}
if ((y>=16) && (y<=120)) // Lower row
{

```

```

if ((x>=0) && (x<=100)) // Button: CLEAR
{
    waitForIt(0, 160,448, 240);
    if(stCurrentLen>0) //jika nilai lebih besar dari 0 atau ada inputan angka lebih
dari 0
    {
        stCurrent[stCurrentLen-1]='\0'; //kosongkan
        asterisc[stCurrentLen-1]='\0'; // kosongkan
        asterisc_partial[stCurrentLen-1]='\0'; //kosongkan
        stCurrentLen=stCurrentLen-1; //setiap di hapus kurangi 1
        clearsmalldisplay(); //hapus lcd
        myGLCD.setColor(0, 0, 0); //hitam
        myGLCD.setBackColor (255, 255, 255);
        myGLCD.print("_____", CENTER, 290); //FIX
        //myGLCD.print(asterisc, 56, 20);
        myGLCD.print(asterisc, 120, 290); //geser kanan angka pwd //FIX
    }
    else
    {
        stCurrent[0]='\0';
        if (setoldcode_flag) //set kode lama
        {
            setoldcode_flag = 0;
            setnewcode_flag = 0;
            repeat_flag = 1;
            cleardisplay();
            blinkmsg("EXITING SETUP", "BLACK");
            msgCurrent = "ENTER PASSCODE";
            printmsg(msgCurrent, "BLACK");
        }
        else
        {
            if (setnewcode_flag) //kode baru
            {
                if (repeat_flag) //fix
                {
                    setoldcode_flag = 0;
                    setnewcode_flag = 0;
                    repeat_flag = 1;
                    cleardisplay();
                    blinkmsg("EXITING SETUP", "BLACK");
                    msgCurrent = "ENTER PASSCODE";
                    printmsg(msgCurrent, "BLACK");
                }
            }
            else
            {
                setoldcode_flag = 0;
                setnewcode_flag = 1;
            }
        }
    }
}

```

```

repeat_flag = 1;
cleardisplay();
msgCurrent = "NEW PASSCODE"; // wajib pakai kode yang baru
printmsg(msgCurrent, "BLACK");
}
}
else
{
cleardisplay();
msgCurrent = "ENTER PASSCODE"; // wajib kode baru
printmsg(msgCurrent, "BLACK");
}
}
}
}
if ((x>=110) && (x<=180)) // Button: ENTER
{
waitForIt(450, 160, 460, 240);
count++; // jika salah maka counter bertambah 1, jika sudah 3 nyalakan buzzer
thisByte=atoi(stCurrent);//array to int
stLastserial=thisByte+48;
Serial.print("data="); // print as an ASCII-encoded decimal - same as "DEC"
Serial.print(stCurrent);
Serial.print("\t"); // prints a tab
Serial.print("DEC=");
Serial.println(stLastserial); // print as an ASCII-encoded decimal
Serial.print("\t"); // prints a tab
Serial.print("HEX=");
Serial.println(stLastserial, HEX); // print as an ASCII-encoded hexadecimal
Serial.print("\t"); // prints a tab
Serial.print("OCT=");
Serial.println(stLastserial, OCT); // print as an ASCII-encoded octal
Serial.print("\t"); // prints a tab
Serial.print("BIN=");
Serial.println(stLastserial, BIN);
if (stCurrentLen==5) //batas 5 ikok pwd
{
stLast = stCurrent; //masukkan angka inputan kedalam memori pwd
stCurrent[0]='\0';
asterisc[0]='\0';
asterisc_partial[0]='\0';
stCurrentLen=0;
}
// print as an ASCII-encoded binary
if (stLast != "00000") //jika memori pwd tidak sama dengan 00000m maka
wajib ganti
kode baru
{

```

```

    if (setnewcode_flag) //jika sudah harus ganti maka wajib kita menggunakan
pwb yang
    baru
    {
    if (repeat_flag)
    {
    stTemp = stLast; //masukkan pwb baru kedalam memori pwd tetap tapi yang
baru
    repeat_flag = 0;
    cleardisplay();
    msgCurrent = "REPEAT PASSCODE";
    printmsg(msgCurrent, "BLACK");
    }
    else
    {
    if (stLast == stTemp) //kita sudah pakai kode baru
    {
    stCode = stLast;
    setoldcode_flag = 0;
    setnewcode_flag = 0;
    repeat_flag = 1;
    cleardisplay();
    blinkmsg("CODE UPDATED!", "GREEN");
    msgCurrent = "ENTER PASSCODE";
    printmsg(msgCurrent, "BLACK");
    }
    else
    {
    repeat_flag = 1; //jika konfirmasi kode baru salah maka tampilkan
cleardisplay();
    blinkmsg("CODE MISMATCH", "RED");
    msgCurrent = "NEW PASSCODE";
    printmsg(msgCurrent, "BLACK");
    }
    }
    }
    else
    {
    if (stLast == stCode)
    {
    if (setoldcode_flag)
    {
    setoldcode_flag = 0;
    setnewcode_flag = 1;
    cleardisplay();
    msgCurrent = "NEW PASSCODE"; // masukkan kode baru ataupun lama
    printmsg(msgCurrent, "BLACK");
    }

```

```

else
{
if (lock_flag)
{
cleardisplay();
blinkmsg("CONTROL LOCKED", "GREEN"); //
lock_flag = 0;
analogWrite(kanan1, 0);
analogWrite(kiri1, 0);
digitalWrite(LR, HIGH);
digitalWrite(RR, HIGH);
digitalWrite(power, LOW);
}
else
{
cleardisplay();
blinkmsg("True Passcode", "GREEN"); //buka gerbang
lock_flag = 1;
count = 0;
digitalWrite(power, HIGH);
}
cleardisplay();
msgCurrent = "ENTER PASSCODE"; //permintaan masukkan kode lagi setelah
masukkan kode
printmsg(msgCurrent, "BLACK");
}
}
else
{
cleardisplay();
if (lock_flag)
blinkmsg("ACCESS DENIED", "RED");
else
blinkmsg("WRONG PASSCODE", "RED"); //salah pwd
printmsg(msgCurrent, "BLACK");
}
}
}
else
{
setoldcode_flag = 1; //blok ngisi pwd baru dari yang lama
setnewcode_flag = 0;
repeat_flag = 1;
cleardisplay();
blinkmsg("ENTERING SETUP", "BLACK");
msgCurrent = "OLD PASSCODE";
printmsg(msgCurrent, "BLACK");
}
}

```

```
}  
else  
{  
  cleardisplay();  
  blinkmsg("TOO SHORT", "RED"); //jangan pendek coyyy  
  stCurrent[0]='\0';  
  asterisc[0]='\0';  
  asterisc_partial[0]='\0';  
  stCurrentLen=0;  
  cleardisplay();  
  printmsg(msgCurrent, "BLACK");  
}  
}  
}  
}  
// read the value from the sensor:  
sensorValue = analogRead(sensorPin); //masukkan nilai adc ke serail  
//Serial.println(sensorValue); // print ADC value of analog reading  
}
```