

## LAMPIRAN

Biaya pembuatan alat

Alat-alat	Harga
Arduino UNO	58.000
RFID reader	14.200
Solenoid lock	55.000
Number pad	8.000
Relay	9.500
Kabel jumper	25.000
ESP-01	50.000
LCD dengan I2c backpack	15.000
Adaptor 12volt	35.000
Lain-lain	50.000
Jumlah	319.700

Code program alat

```

#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <EEPROM.h>
#include <Wire.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

#define SS_PIN 10
#define RST_PIN 9

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key mkey;
char d,d1,d2,d3,d10,d11,d12,d13;
int c,cpa>tag;
bool cp,np,ok,m;
byte nuidPICC[4];

const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}

```

```

};
byte rowPins[ROWS] = {2,3,4,5};
byte colPins[COLS] = {6,7,8};

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

```

```

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

  pinMode(14, OUTPUT);
  lcd.init();
  lcd.backlight();
  lcd.print("SCAN YOUR CARD");
  lcd.setCursor(1, 1);

  SPI.begin();
  rfid.PCD_Init();

  for (byte i = 0; i < 6; i++) {
    mkey.keyByte[i] = 0xFF;
  }

  Serial.println(F("This code scan the MIFARE Classic NUID."));
  if(EEPROM.read(0)>1|EEPROM.read(0)<1){
    EEPROM.update(0, 1);
    EEPROM.update(1, '1');
    EEPROM.update(2, '2');
    EEPROM.update(3, '3');
    EEPROM.update(4, '4');
  }
  d= EEPROM.read(1);
  d1= EEPROM.read(2);
  d2= EEPROM.read(3);
  d3= EEPROM.read(4);

}

void loop() {
  char k = keypad.getKey();

  if (k){
    c++;

```

```

Serial.println(k);

lcd.print("*");
if(c==1){
  d10=k;
}
if(c==2){
  d11=k;
}
if(c==3){
  d12=k;
}
if(c==4){
  d13=k;
}
}
}
if (k=='#'){m=c=cp=np=tag=0; lcd.clear(); lcd.print("SCAN YOUR CARD");}

if(c==4&np==0){
  c=0;
  if(d10=='*&d11=='0'&d12=='0'){
    m=1;
    lcd.clear(); lcd.print("ENTER PASSWORD");
    switch(d13){
      case '0':
        cp=1;

        break;
      case '1':

        tag=1;
        break;
      case '2':
        tag=2;
        break;
      case '3':
        tag=3;
        break;
      case '4':
        tag=4;
        break;
      case '5':
        tag=5;
        break;
    }
  }
}

```

```

    }
    d10=d11=d12=d13=0;
  }
  if(d==d10&d1==d11&d2==d12&d3==d13){
    if(cp==0&m==0){
      lcd.setCursor(0, 1);
      lcd.print("PASSWORD ACCEPTED");
      digitalWrite(14,HIGH);
      delay(3000);
      digitalWrite(14,LOW);
      d10=d11=d12=d13=0;
      lcd.setCursor(0, 1);
      lcd.print(" ");
      lcd.setCursor(0, 1);
    }if(cp==1&tag==0){
      lcd.clear();
      lcd.print("NEW PASSWORD");
      lcd.setCursor(1, 1);
      np=1;
    }
    if(tag>0){

      lcd.clear();
      lcd.print("SCAN ID TAG # ");
      lcd.print(tag);
      lcd.setCursor(0, 1);
    }
    m=0; } else {
    if(!m){ lcd.setCursor(0, 1);
    lcd.print("WRONG PASSWORD");delay(3000);}

    d10=d11=d12=d13=0;
    lcd.setCursor(0, 1);
    lcd.print(" ");
    lcd.setCursor(0, 1);
  }
}

if(c==4&np==1){
  m=0;
  d=d10;
  d1=d11;
  d2=d12;
  d3=d13;
  EEPROM.update(1, d);
  EEPROM.update(2, d1);

```

```

EEPROM.update(3, d2);
EEPROM.update(4, d3);
np=0;
cp=0;
c=0;
d10=d11=d12=d13=0;
lcd.clear(); lcd.print("SCAN YOUR CARD");
lcd.setCursor(0, 1);
}

```

```

if ( ! rfid.PICC_IsNewCardPresent()
return;

```

```

if ( ! rfid.PICC_ReadCardSerial()
return;
printDec(rfid.uid.uidByte, rfid.uid.size);
Serial.println();

```

```

if(tag>0){
  switch(tag){
    case 1:
      EEPROM.update(5, rfid.uid.uidByte[0] );
      EEPROM.update(6, rfid.uid.uidByte[1] );
      EEPROM.update(7, rfid.uid.uidByte[2] );
      EEPROM.update(8, rfid.uid.uidByte[3] );
      break;
    case 2:
      EEPROM.update(9, rfid.uid.uidByte[0] );
      EEPROM.update(10, rfid.uid.uidByte[1] );
      EEPROM.update(11, rfid.uid.uidByte[2] );
      EEPROM.update(12, rfid.uid.uidByte[3] );
      break;
    case 3:
      EEPROM.update(13, rfid.uid.uidByte[0] );
      EEPROM.update(14, rfid.uid.uidByte[1] );
      EEPROM.update(15, rfid.uid.uidByte[2] );
      EEPROM.update(16, rfid.uid.uidByte[3] );
      break;
    case 4:
      EEPROM.update(17, rfid.uid.uidByte[0] );
      EEPROM.update(18, rfid.uid.uidByte[1] );
      EEPROM.update(19, rfid.uid.uidByte[2] );
      EEPROM.update(20, rfid.uid.uidByte[3] );
      break;
    case 5:
      EEPROM.update(21, rfid.uid.uidByte[0] );
      EEPROM.update(22, rfid.uid.uidByte[1] );
      EEPROM.update(23, rfid.uid.uidByte[2] );

```

```

EEPROM.update(24, rfid.uid.uidByte[3] );
    break;

    } tag=0;d10=d11=d12=d13=0;
    lcd.setCursor(0, 1);
    lcd.print(" CARD ID SAVED ");

delay(2000);
    lcd.clear();
    lcd.print("SCAN YOUR CARD");

    cp=0;
    c=0;
    }else for (int i = 1; i < 6; i++){
if (rfid.uid.uidByte[0] == EEPROM.read((i*4)+1)&
rfid.uid.uidByte[1] == EEPROM.read((i*4)+2) &
rfid.uid.uidByte[2] == EEPROM.read((i*4)+3) &
rfid.uid.uidByte[3] == EEPROM.read((i*4)+4) ) ok=1;}

    if(ok==1){
        ok=0;
        Serial.println(F("A new card has been detected."));
        lcd.setCursor(1, 1);
        lcd.print("CARD ACCEPTED ");
        digitalWrite(14,HIGH);
        delay(3000);
        lcd.setCursor(0, 1);
        lcd.print(" ");
        digitalWrite(14,LOW);
        return;
    }else {
        lcd.setCursor(0, 1);
        lcd.print(" ACCESS DENIED ");

delay(2000);
        lcd.setCursor(0, 1);
        lcd.print(" ");

    }

rfid.PICC_HaltA();

rfid.PCD_StopCrypto1();

}

void printDec(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {

```

```
Serial.print(buffer[i] < 0x10 ? " 0" : " ");  
Serial.print(buffer[i], DEC);  
}  
}
```