LAMPIRAN

CODING PROGRAM UTAMA

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

SoftwareSerial mySerial(3, 2);
SoftwareSerial NodeMCU(4, 5);

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);

#define I2C_ADDR 0x27 // <<- Add your address here.
#define Rs_pin 0
#define Rw_pin 1
#define En_pin 2
#define BACKLIGHT_PIN 3
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

LiquidCrystal_I2C lcd(0x27, 16, 4);

uint8_t id;
String inputString = "";
int num = 0;

void setup() {
  // put your setup code here, to run
  once: Serial.begin(115200);
  NodeMCU.begin(115200);
  Wire.begin();
```

```
  // set the data rate for the sensor serial
  port finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }
  finger.getTemplateCount();
  Serial.print("Sensor contains ");
Serial.print(finger.templateCount); Serial.println("
templates");
  Serial.println("Waiting for valid finger...");

  pinMode(8, OUTPUT);
  digitalWrite(8,HIGH);

  lcd.begin ();
  // LCD Backlight ON

  lcd.setBacklight(HIGH);

  lcd.home (); // go home on LCD
  lcd.setCursor (0,0);
  lcd.print("Door Lock System");
  lcd.setCursor (0,1);
  lcd.print("=TeknikKomputer=");
  delay(1000);
//  lcd.clear();
}

void loop() {
  // put your main code here, to run
  repeatedly: lcd.setCursor (0,0);
```

```
  lcd.print(" Silahkan Scan  ");
  lcd.setCursor (0,1);
  lcd.print("Sidik Jari Anda ");

  getFingerprintIDez();
  serialEvent();



//  id = readnumber();
//  if (id == 0) {// ID #0 not allowed, try again!
//     return;
//  }
//  Serial.print("Enrolling ID #");
//  Serial.println(id);
//
//  while (!  getFingerprintEnroll() );
}
```

Coding Program Pengambilan Fingerprint

```
int getFingerprintIDez()
{
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p!=2){
    Serial.println("no finger");
    lcd.clear();
    lcd.setCursor (0,0);
    lcd.print("Sidik Jari Tidak");
    lcd.setCursor (0,1);
    lcd.print("   Terdeteksi   ");
    delay(2000);
  }
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

//  pass = true;
//sidik = false;
  digitalWrite(8,LOW);
  lcd.clear(); lcd.setCursor
  (0,0); lcd.print("Door Lock
  System"); lcd.setCursor
  (0,1);
  lcd.print("=TeknikKomputer=")
  ; lcd.setCursor (0,2);
  lcd.print("Door Open");

  delay(5000);
  digitalWrite(8,HIGH);
```

```
  lcd.setCursor (0,0);
  lcd.print("Door Lock System");
  lcd.setCursor (0,1);
  lcd.print("=TeknikKomputer=");
  lcd.setCursor (0,2);
  lcd.print("Door Close");
  delay(1000);
  lcd.clear();
  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of ");
Serial.println(finger.confidence);
  //Serial.print(finger.fingerID);

  int id = finger.fingerID;
  if (id == 0){
     NodeMCU.write("0");
  }
  else if (id == 1){
     NodeMCU.write("1");
  }
  else if (id == 2){
     NodeMCU.write("2");
  }
  else if (id == 3){
     NodeMCU.write("3");
  }
  else if (id == 4){
     NodeMCU.write("4");
  }
  else if (id == 5){
     NodeMCU.write("5");
  }
  else if (id == 6){
```

```
      NodeMCU.write("6");
  }
  else if (id == 7){
      NodeMCU.write("7");
  }
  else if (id == 8){
      NodeMCU.write("8");
  }
   else if (id == 9){
      NodeMCU.write("9");
  }
   else if (id == 10){
      NodeMCU.write("10");
  }

  Serial.println(",");
  return finger.fingerID;
}

uint8_t readnumber(void) {
  uint8_t num = 0;

  while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
  }
  return num;
}

uint8_t getFingerprintEnroll() {
  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #");
Serial.println(id);
```

```
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.println(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image
    converted"); break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too
    messy"); return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
```

```
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint
    features"); return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint
    features"); return p;
  default:
    Serial.println("Unknown error");
    return p;
}
Serial.println("Remove finger");
delay(2000);

p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
Serial.print("ID ");
Serial.println(id); p = -1;
Serial.println("Place same finger again");

while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
```

```
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
}

// OK success!
p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image
    converted"); break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too
    messy"); return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint
    features"); return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint
    features"); return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
Serial.print("Creating model for #");  Serial.println(id);
```

```
  p = finger.createModel();
  if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR)
    { Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not
    match"); return p;
  } else { Serial.println("Unknown
    error"); return p;

  }
  Serial.print("ID "); Serial.println(id);
  p = finger.storeModel(id); if (p ==
  FINGERPRINT_OK) {
    Serial.println("Stored!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR)
    { Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that
    location"); return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to
    flash"); return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }
  delay(2000);
  id = 0;
}
```