CODING RFID WRITER

```
// ------------[ LIBRARIES ]------------
#include <SPI.h>
#include <MFRC522.h>
#include <String.h>


// ------------[ DEFINE PIN ]------------
#define SS_PIN           10
#define RST_PIN          9
#define RED_PIN          A1
#define GREEN_PIN        A2
#define BLOCK_LENGTH     16
#define BLOCK_LENGTH_EXT  18


// ------------[ OBJECTS AND VARIABLES ]------------
MFRC522 mfrc522 = MFRC522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;


struct inputInformation{
  byte test_read[5][BLOCK_LENGTH_EXT];
  byte auth[BLOCK_LENGTH];
  byte nim[BLOCK_LENGTH];
  byte ttl[BLOCK_LENGTH];
  byte no_hp[BLOCK_LENGTH];
  byte auth_message[BLOCK_LENGTH];
}card_content;


enum locationBlock{
  AUTH = 2,
  NIM = 4,
  TTL = 5,
  NO_HP = 6
};
String temp_str;
byte buffer_ATQA[10], buffer_size = sizeof(buffer_ATQA), report;
```

```
// ------------[ FUNCTIONS ]------------
int writeCard(int block_number, byte arr[]){


  // 4th block starting from 0 is forbidden, error message 0
  int forbidden_block = int(block_number/4*4) + 3;
  if(block_number > 0 && (block_number+1)%4==0)
     return 0;


  // authentication status for the card, error message 1
  byte status = mfrc522.PCD_Authenticate(
    MFRC522::PICC_CMD_MF_AUTH_KEY_A,
    forbidden_block, &key, &(mfrc522.uid)
  );
  if(status != MFRC522::STATUS_OK)
     return 1;


  // write card at the desired block_number, error message 2
  status = mfrc522.MIFARE_Write(block_number, arr, BLOCK_LENGTH);
  if(status != MFRC522::STATUS_OK)
     return 2;


  // if success, return 3
  return 3;
}


int readCard(int block_number, byte arr[]){



  // 4th block starting from 0 is forbidden
  int forbidden_block = int(block_number/4*4) + 3;


  // authentication status for the card, error message 4
  byte status = mfrc522.PCD_Authenticate(
```

```
    MFRC522::PICC_CMD_MF_AUTH_KEY_A,
    forbidden_block, &key, &(mfrc522.uid)
  );
  if(status != MFRC522::STATUS_OK)
     return 4;


  // define the buffersize (+2 for error correcting bit)
  byte buffersize = BLOCK_LENGTH_EXT;


  // read the card at the desired block number, error message 5
  status = mfrc522.MIFARE_Read(block_number, arr, &buffersize);
  if(status != MFRC522::STATUS_OK)
     return 5;


  // if success, return 6
  return 6;
}


void printCard(byte arr[]){

  for(int i=0; i<BLOCK_LENGTH; ++i){
     Serial.write(arr[i]);
  }
  Serial.println();
}



void strToByte(String str, byte x[]){

  for(int i=0; i<BLOCK_LENGTH; ++i){
     x[i] = byte(str[i]);
  }
}
```

```
bool compareByte(byte data_1[], byte data_2[]){

  for(int i=0; i<BLOCK_LENGTH; ++i){
    if(data_1[i] != data_2[i])
      break;
    if(i == BLOCK_LENGTH - 1)
      return true;
  }
  return false;
}


void updateCardContent(char str[]){

  // iterate to all the string content
  bool nim = true, ttl = false, no_hp = false;
  int i = 0, j = 0;
  while(str[i] != '$'){

    // catchers
    if(str[i] == '@'){
      nim = false;
      ttl = true;
      ++i; j = 0;
    }

    if(str[i] == '#'){
      ttl = false;
      no_hp = true;
      ++i; j = 0;
    }

    // inputing the content
    if(nim){
      card_content.nim[j] = byte(str[i]);
      ++j;
    }

    else if(ttl){
      card_content.ttl[j] = byte(str[i]);
```

```
    ++j;
   }

   else if(no_hp){
    card_content.no_hp[j] = byte(str[i]);
    ++j;
   }
   Serial.println(i);
   ++i;
  }

}


void(* resetFunc)(void) = 0;


// ------------[ SETUP ]------------
void setup(){


  // open serial and SPI
  Serial.begin(9600);
  SPI.begin();


  // initialize LED and RFID writer & reader
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  mfrc522.PCD_Init();
  digitalWrite(RED_PIN, HIGH);


  // initialize 6-bytes key of MIFARE::key
  for(byte i=0; i<6; ++i){
   key.keyByte[i] = 0xFF;
  }


  // define authentication message for writing the card
```

```
  temp_str = "unwritable000000";
  strToByte(temp_str, card_content.auth_message);
}


// ------------[ LOOP ]------------
void loop(){


  // serial print string of 0, for python detection
  Serial.println("0");


  // check whether a new card exists
  if(!mfrc522.PICC_IsNewCardPresent()){


    // turn on the red LED
    digitalWrite(RED_PIN, HIGH);
    digitalWrite(GREEN_PIN, LOW);
    return;
  }

  // check whether the new card valid
  if(!mfrc522.PICC_ReadCardSerial()){

    // removing the card mechanism
    while(true){

      // blink the LED with red
      digitalWrite(RED_PIN, LOW); delay(400);
      digitalWrite(RED_PIN, HIGH); delay(400);
      Serial.println("0");
      report = readCard(13, card_content.test_read[4]);
      if(report != 6){

        // halting the RFID process
        delay(200); mfrc522.PICC_HaltA(); delay(200);
        report = mfrc522.PICC_WakeupA(buffer_ATQA, &buffer_size);
        report = mfrc522.PICC_RequestA(buffer_ATQA, &buffer_size);
```

```
      resetFunc();
      return;
    }
  }
}


// if the existing new card valid
else{
    // turn on the green
    digitalWrite(RED_PIN, LOW);
    digitalWrite(GREEN_PIN, HIGH);

    // print string of 1, for python detection
    Serial.println("1");
}


// read for NIM, TTL, and NO_HP from python
while(Serial.available()==0){}
String python_str = Serial.readStringUntil('\n');
char python_buffer[64];
python_str.toCharArray(python_buffer, 64);
updateCardContent(python_buffer);


// write the inputted data from python to card
writeCard(AUTH, card_content.auth_message);
writeCard(NIM, card_content.nim);
writeCard(TTL, card_content.ttl);
report = writeCard(NO_HP, card_content.no_hp);


// info to python
if(report == 3)
  Serial.println("y");
else
  Serial.println("n");
```

```
  // removing the card mechanism
  while(true){

    // turn off the green LED
    digitalWrite(GREEN_PIN, LOW);

    // blink the LED with red
    digitalWrite(RED_PIN, LOW); delay(400);
    digitalWrite(RED_PIN, HIGH); delay(400);
    report = readCard(13, card_content.test_read[4]);
    if(report != 6){

      // halting the RFID process
      delay(200); mfrc522.PICC_HaltA(); delay(200);
      report = mfrc522.PICC_WakeupA(buffer_ATQA, &buffer_size);
      report = mfrc522.PICC_RequestA(buffer_ATQA, &buffer_size);
      resetFunc();
      return;
    }
  }
}
```

CODING ALAT VALIDASI KEANGGOTAAN PERPUSTAKAAN YANG MEMPERHATIKAN PROTOKOL KESEHATAN BERBASIS MIKROKONTROLER

```
// ---------------[ LIBRARIES ]--------------- //
#include <SPI.h>
#include <Wire.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <String.h>
#include "MFRC522.h"
#include "Adafruit_MLX90614.h"
#include "Adafruit_SH110X.h"
#include "Adafruit_GFX.h"


// ---------------[ DEFINITION ]--------------- //
#define SCREEN_WIDTH      128
#define SCREEN_HEIGHT     64
#define OLED_ADDR         0x3C
#define MLX_ADDR          0x5A
#define RST_PIN           17
#define SS_PIN            5
#define OLED_RST          -1
#define BUZZER_PIN        16
#define RED_PIN           33
#define GREEN_PIN         32

#define WIFI_SSID         "WY_Wifi"
#define WIFI_PASS         "ber217an"
#define WIFI_TIMEOUT_MS   8000

#define    FIREBASE_HOST                  "wy-rfid-health-check-default-rtdb.asia-
southeast1.firebasedatabase.app"
#define                                                    FIREBASE_AUTH
"GQUz4pPEpOWdq3Fzfrco8Ndm12DU6QIbEHnS8POh"

//#define SERVER_LOCAL_IP   "192.168.128.52"
//#define SERVER_PORT       10013
//#define SERVER_TIMEOUT    5000
```

```
#define MAX_VISITOR       6
#define MAX_TEMP          37.0

#define BLOCK_LENGTH      16
#define BLOCK_LENGTH_EXT  18

#define DISPLAY_IDLE_MS   5000


// ---------------[ OBJECTS ]---------------- //
Adafruit_SH1106G oled = Adafruit_SH1106G(SCREEN_WIDTH,
SCREEN_HEIGHT, &Wire, OLED_RST);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
MFRC522 mfrc = MFRC522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;


// ---------------[ VARIABLES ]---------------- //
uint16_t visitor_num = 0;
float visitor_temp = 0.;
unsigned long int start_time, idle_time;
byte buffer_ATQA[10], buffer_size = sizeof(buffer_ATQA), report;

struct cardInformation{
  byte auth[BLOCK_LENGTH_EXT];
  byte nim[BLOCK_LENGTH_EXT];
  byte ttl[BLOCK_LENGTH_EXT];
  byte no_hp[BLOCK_LENGTH_EXT];
}card_content;

enum locationBlock{
  AUTH = 2,
  NIM = 4,
  TTL = 5,
  NO_HP = 6
};
```

// ---------------[BITMAP IMAGES]--------------- //

```
const unsigned char WYlogo [] PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x7f, 0xc0, 0x7f, 0xe0, 0x00, 0x03, 0xfe, 0x3f, 0xe0, 0x00, 0x03,
0xfe, 0x00, 0x00,
  0x00, 0x00, 0x7f, 0xe0, 0x7f, 0xf0, 0x00, 0x03, 0xfe, 0x3f, 0xf0, 0x00, 0x07,
0xfe, 0x00, 0x00,
  0x00, 0x00, 0x7f, 0xf0, 0x7f, 0xf8, 0x00, 0x03, 0xfe, 0x3f, 0xf8, 0x00, 0x0f,
0xfe, 0x00, 0x00,
  0x00, 0x00, 0x7f, 0xf8, 0x7f, 0xf8, 0x00, 0x03, 0xfe, 0x3f, 0xfc, 0x00, 0x0f,
0xfe, 0x00, 0x00,
  0x00, 0x00, 0x7f, 0xfc, 0x3f, 0xfc, 0x00, 0x03, 0xfe, 0x3f, 0xfe, 0x00, 0x1f,
0xfe, 0x00, 0x00,
  0x00, 0x00, 0x3f, 0xfe, 0x3f, 0xfe, 0x00, 0x03, 0xfe, 0x1f, 0xfe, 0x00, 0x3f,
0xfe, 0x00, 0x00,
  0x00, 0x00, 0x1f, 0xfe, 0x1f, 0xff, 0x00, 0x03, 0xfe, 0x0f, 0xff, 0x00, 0x7f,
0xfc, 0x00, 0x00,
```

0x00, 0x00, 0x0f, 0xff, 0x0f, 0xff, 0x80, 0x03, 0xfe, 0x0f, 0xff, 0x80, 0xff, 0xf8, 0x00, 0x00,
0x00, 0x00, 0x0f, 0xff, 0x87, 0xff, 0xc0, 0x03, 0xfe, 0x07, 0xff, 0xc1, 0xff, 0xf0, 0x00, 0x00,
0x00, 0x00, 0x07, 0xff, 0xc3, 0xff, 0xe0, 0x03, 0xfe, 0x03, 0xff, 0xc3, 0xff, 0xe0, 0x00, 0x00,
0x00, 0x00, 0x03, 0xff, 0xe1, 0xff, 0xe0, 0x03, 0xfe, 0x01, 0xff, 0xc7, 0xff, 0xc0, 0x00, 0x00,
0x00, 0x00, 0x01, 0xff, 0xf0, 0xff, 0xf0, 0x03, 0xfe, 0x00, 0xff, 0x87, 0xff, 0x80, 0x00, 0x00,
0x00, 0x00, 0x00, 0xff, 0xf8, 0xff, 0xf8, 0x03, 0xfe, 0x00, 0x7f, 0x0f, 0xff, 0x80, 0x00, 0x00,
0x00, 0x00, 0x00, 0x7f, 0xfc, 0x7f, 0xfc, 0x03, 0xfe, 0x00, 0x3e, 0x1f, 0xff, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x3f, 0xfc, 0x3f, 0xfe, 0x03, 0xfe, 0x00, 0x3c, 0x3f, 0xfe, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x3f, 0xfe, 0x1f, 0xff, 0x03, 0xfe, 0x00, 0x18, 0x7f, 0xfc, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x1f, 0xff, 0x0f, 0xff, 0x83, 0xfe, 0x00, 0x00, 0xff, 0xf8, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0f, 0xff, 0x87, 0xff, 0xc3, 0xfe, 0x00, 0x01, 0xff, 0xf0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x07, 0xff, 0xc3, 0xff, 0xc3, 0xfe, 0x00, 0x03, 0xff, 0xe0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x03, 0xff, 0xe3, 0xff, 0xe1, 0xfe, 0x00, 0x03, 0xff, 0xc0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x01, 0xff, 0xf1, 0xff, 0xf0, 0xfe, 0x00, 0x03, 0xff, 0xc0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xff, 0xf8, 0xff, 0xf8, 0x7e, 0x00, 0x03, 0xff, 0x80, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7f, 0xfc, 0x7f, 0xfc, 0x3e, 0x00, 0x03, 0xff, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7f, 0xfc, 0x3f, 0xfe, 0x3e, 0x00, 0x03, 0xfe, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3f, 0xfe, 0x1f, 0xff, 0x1e, 0x00, 0x03, 0xfe, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0x0f, 0xff, 0x8e, 0x00, 0x03, 0xfe, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0x87, 0xff, 0xc6, 0x00, 0x03, 0xfe, 0x00, 0x00, 0x00, 0x00,

```
  0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xc7, 0xff, 0xc2, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xe3, 0xff, 0xe0, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xf1, 0xff, 0xf0, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xf8, 0xff, 0xf8, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xf8, 0x7f, 0xfc, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xfc, 0x3f, 0xfe, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xfe, 0x1f, 0xfe, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xfe, 0x0f, 0xfe, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xfe, 0x07, 0xfe, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xfe, 0x07, 0xfe, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xfc, 0x03, 0xfe, 0x00, 0x03, 0xfe, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
```

```
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00
};


// ---------------[ FUNCTIONS ]--------------- //
float getVisitorTemp(){

  // take the mean of 30 temperature samples
  float sum = 0.;
  for(int i = 0; i < 30; ++i){
    sum += (float)mlx.readObjectTempC();
    delay(15);
  }
  return sum/30.;
}


int readCard(int block_number, byte arr[]){


  // 4th block starting from 0 is forbidden
  int forbidden_block = int(block_number/4*4) + 3;


  // authentication status for the card, error message 4
  byte status = mfrc.PCD_Authenticate(
    MFRC522::PICC_CMD_MF_AUTH_KEY_A,
    forbidden_block, &key, &(mfrc.uid)
  );
  if(status != MFRC522::STATUS_OK)
      return 4;
```

```
  // define the buffersize (+2 for error correcting bit)
  byte buffersize = BLOCK_LENGTH_EXT;


  // read the card at the desired block number, error message 5
  status = mfrc.MIFARE_Read(block_number, arr, &buffersize);
  if(status != MFRC522::STATUS_OK)
     return 5;


  // if success, return 6
  return 6;
}


void strToByte(String str, byte x[]){

  for(int i=0; i<BLOCK_LENGTH; ++i){
     x[i] = byte(str[i]);
  }
}


bool compareByte(byte data_1[], byte data_2[]){

  for(int i=0; i<BLOCK_LENGTH; ++i){
    if(data_1[i] != data_2[i])
      break;
    if(i == BLOCK_LENGTH - 1)
      return true;
  }
  return false;
}


void(* resetFunc)(void) = 0;
```

```
// ---------------[ SETUP ]--------------- //
void setup(){


  // start up delay
  delay(250);
  start_time = millis();


  // initialize Serial, SPI, OLED, MLX90614, MFRC522, WIFI, buzzer, and LEDs
  Serial.begin(115200);
  SPI.begin();
  oled.begin(OLED_ADDR, true);
  mlx.begin();
  mfrc.PCD_Init();
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(RED_PIN, OUTPUT);


  // initialize 6-bytes key of MIFARE::key
  for(byte i=0; i<6; ++i){
    key.keyByte[i] = 0xFF;
  }


  // start splash screen on OLED display
  oled.clearDisplay();
  oled.drawBitmap(0, 0, WYlogo, 128, 64, 1);
  oled.display();
  digitalWrite(BUZZER_PIN, HIGH); delay(300);
  digitalWrite(BUZZER_PIN, LOW); delay(300);
  digitalWrite(BUZZER_PIN, HIGH); delay(300);
  digitalWrite(BUZZER_PIN, LOW); delay(300);
  digitalWrite(BUZZER_PIN, HIGH); delay(300);
  digitalWrite(BUZZER_PIN, LOW); delay(300);
```

```
  // connecting to Wifi
  oled.clearDisplay();
  oled.setTextColor(SH110X_WHITE);
  oled.setCursor(0, 0);
  oled.setTextSize(1);
  oled.print("Connecting to "); oled.print(WIFI_SSID); oled.println("\n");
  oled.display();

  while(WiFi.status() != WL_CONNECTED && millis() - start_time < WIFI_TIMEOUT_MS){
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    delay(100);
  }

  if(WiFi.status() != WL_CONNECTED){
    oled.println("Connection failed!");
    oled.println("Please restart the device.");
    oled.display();
    while(true){}
  }

  else{
    oled.println("Connected!");
    oled.println("ESP IP: ");
    oled.print(WiFi.localIP());
    oled.display();
    delay(4000);
  }


  // connecting to database
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);


  // loop start time
  start_time = millis();
}
```

```
// ---------------[ LOOP ]--------------- //
void loop() {


  // count the idle time everytime a loop started
  idle_time = millis() - start_time;


  // OLED print config
  oled.clearDisplay();
  oled.setCursor(0, 0);


  // idle display
  if(idle_time < DISPLAY_IDLE_MS){
    oled.setTextSize(1);
    oled.println("Jumlah pengunjung: \n\n");
    oled.setTextSize(4);
    oled.print("  ");
    oled.print(visitor_num);
    oled.setTextSize(1);
    oled.print("/6");
  }

  else{
    oled.setTextSize(1);
    oled.println("Suhu ruangan: \n\n");
    oled.setTextSize(3);
    oled.print(mlx.readAmbientTempC(), 1);
    oled.print(" C");

    if(idle_time >= 2*DISPLAY_IDLE_MS)
      start_time = millis();
  }

  oled.display();


  // check whether a new card detected or nah
  if(!mfrc.PICC_IsNewCardPresent()){
```

```
  // turn on the red LED
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(GREEN_PIN, LOW);
  return;
}

if(!mfrc.PICC_ReadCardSerial()){
  return;
}


// if the detected card is okay
oled.clearDisplay();
oled.setCursor(5, 25);
oled.setTextSize(2);
oled.print("Memindai.....");
oled.display();


// read the card data
readCard(AUTH, card_content.auth);
readCard(NIM, card_content.nim);
readCard(TTL, card_content.ttl);
readCard(NO_HP, card_content.no_hp);
delay(1000);


// send NIM to server and search inside the database
FirebaseData database;
FirebaseJson json;
String temp_nim = (char*)card_content.nim;
String mahasiswa_directory = "/mahasiswa/" + temp_nim;
String nim_directory = mahasiswa_directory + "/nim";
String isInside_directory = mahasiswa_directory + "/isInside";
bool is_exist = false;
bool is_inside = false;

if(Firebase.getString(database, nim_directory)){
```

```
  is_exist = true;

  if(Firebase.getBool(database, isInside_directory)){

    if(database.boolData())
      is_inside = true;
  }
}


// case check
oled.clearDisplay();
oled.setCursor(0, 0);


// check whether the visitor is from inside or outside
if(!is_inside){

  // is visitor registered?
  if(is_exist){

    // check visitor's temperature
    visitor_temp = getVisitorTemp();
    if(visitor_temp <= MAX_TEMP){

      // check the current visitor's number
      if(visitor_num < MAX_VISITOR){

        // visitor accepted
        ++visitor_num;
        json.set("/isInside", true);
        Firebase.updateNode(database, mahasiswa_directory, json);


        // notification
        oled.setTextSize(1);
        oled.setCursor(0, 0);
        oled.println("Selamat datang, silakan masuk!\n\n");
        oled.setTextSize(2);
        oled.println(visitor_temp);
```

```
  oled.display();

  // blink the green and beep the buzzer
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(300);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  delay(300);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(300);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  delay(3000);
}

else{

  // max capacity
  oled.setTextSize(2);
  oled.setCursor(5, 25);
  oled.println("Kapasitas penuh!");
  oled.display();

  // blink the red and beep the buzzer
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(600);
  digitalWrite(RED_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  delay(600);
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(600);
  digitalWrite(RED_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  delay(600);
  digitalWrite(RED_PIN, HIGH);
```

```
    digitalWrite(BUZZER_PIN, HIGH);
    delay(600);
    digitalWrite(RED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    delay(3000);
   }
  }

  else{

   // visitor's temperature is too hot
   oled.setTextSize(1);
   oled.println("Suhu anda melebihi batas! (> 37.0)\n");
   oled.setTextSize(2);
   oled.print(visitor_temp);
   oled.display();

   // blink the red and beep the buzzer
   digitalWrite(RED_PIN, HIGH);
   digitalWrite(BUZZER_PIN, HIGH);
   delay(600);
   digitalWrite(RED_PIN, LOW);
   digitalWrite(BUZZER_PIN, LOW);
   delay(600);
   digitalWrite(RED_PIN, HIGH);
   digitalWrite(BUZZER_PIN, HIGH);
   delay(600);
   digitalWrite(RED_PIN, LOW);
   digitalWrite(BUZZER_PIN, LOW);
   delay(600);
   digitalWrite(RED_PIN, HIGH);
   digitalWrite(BUZZER_PIN, HIGH);
   delay(600);
   digitalWrite(RED_PIN, LOW);
   digitalWrite(BUZZER_PIN, LOW);
   delay(3000);
  }
}
```

```
    else{

      // visitor not detected inside the database
      oled.setTextSize(1);
      oled.print("Anda belum terdaftar di database! Silakan mendaftar terlebih dahulu.");
      oled.display();

      // blink the red and beep the buzzer
      digitalWrite(RED_PIN, HIGH);
      digitalWrite(BUZZER_PIN, HIGH);
      delay(600);
      digitalWrite(RED_PIN, LOW);
      digitalWrite(BUZZER_PIN, LOW);
      delay(600);
      digitalWrite(RED_PIN, HIGH);
      digitalWrite(BUZZER_PIN, HIGH);
      delay(600);
      digitalWrite(RED_PIN, LOW);
      digitalWrite(BUZZER_PIN, LOW);
      delay(600);
      digitalWrite(RED_PIN, HIGH);
      digitalWrite(BUZZER_PIN, HIGH);
      delay(600);
      digitalWrite(RED_PIN, LOW);
      digitalWrite(BUZZER_PIN, LOW);
      delay(3000);
    }
  }

  else{

    // visitor leaving
    --visitor_num;
    json.set("/isInside", false);
    Firebase.updateNode(database, mahasiswa_directory, json);
    oled.setCursor(5, 25);
    oled.setTextSize(2);
    oled.print("Selamat Jalan!");
    oled.display();
```

```
  // blink the green and beep the buzzer
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(300);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  delay(300);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(300);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  delay(3000);
 }

 delay(200); mfrc.PICC_HaltA(); delay(200);
 report = mfrc.PICC_WakeupA(buffer_ATQA, &buffer_size);
 report = mfrc.PICC_RequestA(buffer_ATQA, &buffer_size);
 mfrc.PCD_Init();
 return;
}
```

CODING PYTHON APLIKASI RFID WRITER

```
# import modules
from tkinter import *
from serial import *
from time import sleep
import threading
import pyrebase


# rfidApp class
class rfidApp:


    # constructor
    def __init__(self) -> None:

        # firebase config
        self.firebaseConfig = {
            "apiKey": "AIzaSyA_PF7r7x3hgfNYb7lupdyMABFpYRdTKgY",
            "authDomain": "wy-rfid-health-check.firebaseapp.com",
                        "databaseURL":    "https://wy-rfid-health-check-default-rtdb.asia-
southeast1.firebasedatabase.app",
            "projectId": "wy-rfid-health-check",
            "storageBucket": "wy-rfid-health-check.appspot.com",
            "messagingSenderId": "940789530083",
            "appId": "1:940789530083:web:fe1705c517edca415534d6",
            "measurementId": "G-6B9PVTQ2FS"
        }
        self.firebase = pyrebase.initialize_app(self.firebaseConfig)
        self.database = self.firebase.database()

        # student dictionary
        self.studentData = {
            "nim": "0",
            "ttl": "0",
            "noHp": "0",
            "isInside": False
        }

        # windows setup
        self.root = Tk()
        self.root.geometry('500x300')
        self.root.resizable(False, False)
        self.root.title(string = "WY RFID Writer")
```

```
self.root.iconbitmap("logo.ico")

# main container frame
self.mainFrame = Frame(self.root)
self.mainFrame.place(
    relheight = 1,
    relwidth = 1,
    relx = 0,
    rely = 0,
    anchor = NW
)

# container for title and buttons
self.titleFrame = Frame(self.mainFrame)
self.titleFrame.place(
    relheight = 0.25,
    relwidth = 1,
    relx = 0,
    rely = 0,
    anchor = NW
)
self.titleLabel = Label(
    self.titleFrame,
    text = "WY RFID Writer",
    font = ("Mechsuit", 16)
)
self.titleLabel.place(
    relx = 0.035,
    rely = 0.35,
    anchor = W
)
self.statusLabel = Label(
    self.titleFrame,
    text = "Not connected",
    font = ("Calibri", 11),
    fg = "red"
)
self.statusLabel.place(
    relx = 0.035,
    rely = 0.8,
    anchor = W
)
self.writeButton = Button(
    self.titleFrame,
    text = "Write",
```

```
    font = ("Calibri", 12),
    state = DISABLED,
    command = lambda:[
        self.writeCard()
    ]
)
self.writeButton.place(
    relheight = 0.5,
    relwidth = 0.2,
    relx = 0.75,
    rely = 0.5,
    anchor = W
)

# container 1: NIM
self.nimFrame = Frame(self.mainFrame)
self.nimFrame.place(
    relheight = 0.23,
    relwidth = 1,
    relx = 0,
    rely = 0.25,
    anchor = NW
)
self.nimLabel = Label(
    self.nimFrame,
    text = "Nomor Induk Mahasiswa (NIM): ",
    font = ("Calibri", 11)
)
self.nimLabel.place(
    relx = 0.035,
    rely = 0.15,
    anchor = NW
)
self.nimEntry = Entry(
    self.nimFrame,
    justify = LEFT,
    font = ("Calibri", 13)
)
self.nimEntry.place(
    relheight = 0.5,
    relwidth = 0.92,
    relx = 0.5,
    rely = 0.5,
    anchor = N
)
```

```
# container 2: TTL
self.ttlFrame = Frame(self.mainFrame)
self.ttlFrame.place(
    relheight = 0.23,
    relwidth = 1,
    relx = 0,
    rely = 0.47,
    anchor = NW
)
self.ttlLabel = Label(
    self.ttlFrame,
    text = "Tempat Tanggal Lahir (ddmmyyyy):",
    font = ("Calibri", 11)
)
self.ttlLabel.place(
    relx = 0.035,
    rely = 0.15,
    anchor = NW
)
self.ttlEntry = Entry(
    self.ttlFrame,
    justify = LEFT,
    font = ("Calibri", 13)
)
self.ttlEntry.place(
    relheight = 0.5,
    relwidth = 0.92,
    relx = 0.5,
    rely = 0.5,
    anchor = N
)

# container 3: NO. HP
self.hpFrame = Frame(self.mainFrame)
self.hpFrame.place(
    relheight = 0.23,
    relwidth = 1,
    relx = 0,
    rely = 0.7,
    anchor = NW
)
self.hpLabel = Label(
    self.hpFrame,
    text = "Nomor Handphone: ",
```

```
        font = ("Calibri", 11)
    )
    self.hpLabel.place(
        relx = 0.035,
        rely = 0.15,
        anchor = NW
    )
    self.hpEntry = Entry(
        self.hpFrame,
        justify = LEFT,
        font = ("Calibri", 13)
    )
    self.hpEntry.place(
        relheight = 0.5,
        relwidth = 0.92,
        relx = 0.5,
        rely = 0.5,
        anchor = N
    )


# arduino port
def portInit(self) -> None:

    # port
    self.port = Serial("COM3", 9600)
    self.port.timeout = 1

    # variables
    self.sync = False
    self.threadStop = False
    self.readyWrite = False


# thread
def commThread(self) -> None:

    # loop
    while True:

        if not self.threadStop:
            try:
                if not self.readyWrite:
                    self.data = self.port.readline().decode('ascii')
                    self.data = self.data[:-2]
```

```
                self.sync = True
                print(self.data)
                self.statusLabel.configure(
                    text = "Connected to RFID writer (COM3)",
                    fg = "green"
                )

            if self.data == "1" and not self.readyWrite:
                self.readyWrite = True
                self.statusLabel.configure(
                    text = "COM3 ready to write!"
                )
                self.writeButton.configure(
                    state = NORMAL
                )
                self.threadStop = True

            if self.data == "x" and not self.readyWrite:
                self.statusLabel.configure(
                    text = "Card has been written before!",
                    fg = "red"
                )

        except:
            self.sync = False
            print("Synchronizing...")
            self.statusLabel.configure(
                text = "Not connected",
                fg = "red"
            )
            self.writeButton.configure(
                state = DISABLED
            )

    else:
        break


# write button function
def writeCard(self) -> None:

    # stop the thread
    self.thread.join()
    del self.thread
```

```
        # getting the entry string
        self.nimString = self.nimEntry.get()
        self.ttlString = self.ttlEntry.get()
        self.hpString = self.hpEntry.get()

        # inputting to student dictionary
        self.studentData["nim"] = self.nimString
        self.studentData["ttl"] = self.ttlString
        self.studentData["noHp"] = self.hpString

        # combine all string
         self.catString = self.nimString + "@" + self.ttlString + "#" + self.hpString +
"$"
        print(self.catString)

        # write the catString to port
        self.port.write(self.catString.encode())
        sleep(0.5)

        # check whether the writing process completed succesfully or not
        while True:
            self.data = self.port.readline().decode('ascii')
            self.data = self.data[:-2]
            print(self.data)
            if self.data == "y" or self.data == "n":
                break

        # if successful
        if self.data == "y":
            self.readyWrite = False
            self.sync = False
            self.threadStop = False

            # push to database
                                                                self.tempData =
self.database.child("mahasiswa").child(self.nimString).get()
            try:
                if self.tempData.val()["nim"] == self.nimString:
                    self.database.child("mahasiswa").child(self.nimString).update(self.stu
dentData)
            except:
                self.database.child("mahasiswa").child(self.nimString).set(self.studentD
ata)

            # configure
```

```
            self.statusLabel.configure(
                text = "Data successfully written to card and database!"
            )
            self.writeButton.configure(
                state = DISABLED
            )

        # if failed
        elif self.data == "n":
            self.readyWrite = False
            self.sync = False
            self.threadStop = False

            self.statusLabel.configure(
                text = "Data failed to be written to the RFID card!",
                fg = "red"
            )
            self.writeButton.configure(
                state = DISABLED
            )

        # sleep
        sleep(3)

        # start the thread again
        self.thread = threading.Thread(target = self.commThread)
        self.thread.start()


    # launch
    def launch(self) -> None:

        # port communication start
        self.portInit()
        self.thread = threading.Thread(target = self.commThread)
        self.thread.start()

        # mainloop start
        self.root.mainloop()


# main process
app = rfidApp()
app.launch()
```

CODING PYTHON DATABASE

```
# import modules
import socket, time, pyrebase


# database class
class databaseControl:


    # constructor
    def __init__(self) -> None:

        # firebase config
        self.firebaseConfig = {
            "apiKey": "AIzaSyA_PF7r7x3hgfNYb7lupdyMABFpYRdTKgY",
            "authDomain": "wy-rfid-health-check.firebaseapp.com",
                        "databaseURL":    "https://wy-rfid-health-check-default-rtdb.asia-
southeast1.firebasedatabase.app",
            "projectId": "wy-rfid-health-check",
            "storageBucket": "wy-rfid-health-check.appspot.com",
            "messagingSenderId": "940789530083",
            "appId": "1:940789530083:web:fe1705c517edca415534d6",
            "measurementId": "G-6B9PVTQ2FS"
        }
        self.firebase = pyrebase.initialize_app(self.firebaseConfig)
        self.database = self.firebase.database()

        # socket start
        self.sock = socket.socket()
        self.sock.bind(('0.0.0.0', 10013))


    # func for searching nim
    def searchForNim(self, str_val):

        # search in database
                                                                student                =
self.database.child("mahasiswa").order_by_child("nim").equal_to(str_val).get()

        if len(student) != 0:
            return True

        else:
            return False
```

```
    # start loop
    def launch(self) -> None:

        print("Waiting for a connection request...")
        self.sock.listen(0)

        while True:

            print("Waiting for client...")
            self.client, self.addr = self.sock.accept()

            while True:
                temp = self.client.recv(32)

                if len(temp) != 0:
                    self.msg = str(temp.decode('UTF-8'))

                    if self.searchForNim(self.msg):
                        time.sleep(1)
                        self.client.sendall(bytes("1", 'UTF-8'))

                    else:
                        time.sleep(1)
                        self.client.sendall(bytes("0", 'UTF-8'))

                else:
                    break

            self.client.close()


# main process
db = databaseControl()
db.launch()
```

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

LEMBAR KONSULTASI LAPORAN AKHIR

Nama : M Willy Kurniawan
Nim : 061930701649
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer
Dosen Pembimbing I : Hartati Deviana, S.Kom., M.Kom.

No.	Tanggal	Uraian Bimbingan	Keterangan	Paraf Pembimbing
		ACC JUDUL LAPORAN AKHIR		
		PERBAIKAN PENULISAN		
		PERBAIKAN BAB I - III		
		PERBAIKAN BAB III		
		PERBAIKAN BAB IV		
		ACC BAB I - III		
		ACC LAPORAN AKHIR		

Palembang, ... 2022
Mengetahui,
Ketua Jurusan Teknik Komputer

Azwardi, S.T.,M.T
NIP 19700523200501004

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

LEMBAR KONSULTASI LAPORAN AKHIR

Nama : M Willy Kurniawan
Nim : 061930701649
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer
Dosen Pembimbing II : Indarto, S.T., M.CS.

No.	Tanggal	Uraian Bimbingan	Keterangan	Paraf Pembimbing
	29-7-2022	Penulisan yang rapi. Maksud dari judul dan manfaatnya.		
	1-8-2022	Perbaikan BAB I.		
	2-8-2022	Perbaikan BAB II.		
	4-8-2022	Perbaikan BAB II.		
	5-8-2022	Perbaikan BAB III.		
	7-8-2022	ACC BAB I-III.		
	8-8-2022	Perbaikan BAB IV.		
	8-8-2022	ACC LAPORAN AKHIR.		

Palembang, ... 2022
Mengetahui,
Ketua Jurusan Teknik Komputer

Azwardi, S.T.,M.T
NIP 19700523200501004

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polisri.ac.id E-mail : info@polisri.ac.id

REKOMENDASI SEMINAR LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama	:	M. Willy Kurniawan
NIM	:	061930701649
Jurusan/Program Studi	:	Teknik Komputer / D3 Teknik Komputer
Judul Laporan	:	Rancang Bangun Alat Validasi Keanggotaan Perpustakaan Yang Memperhatikan Protokol Kesehatan Berbasis Mikrokontroler

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Seminar Laporan Akhir pada Tahun Akademik 2021/2022.

Palembang, 8 Agustus 2022

Pembimbing I,

Hartati Deviana, S.Kom., M.Kom.
NIP. 197405262008122001

Pembimbing II,

Indarto, S.T., M.Cs.
NIP. 197307062005011003

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI TUGAS AKHIR (TA)	

Ruang : 2

Dosen Penguji : Emma Laila,S.Kom.,M.Kom.

Nama Mahasiswa : M. Willy Kurniawan

NIM : 061930701649

Jurusan / Program Studi : Teknik Komputer / D3 Teknik Komputer

Judul LA / Skripsi : Rancang Bangun Alat Validasi Keanggotaan Perpustakaan Yang Memperhatikan Protokol Kesehatan Berbasis Mikrokontroler

No	Uraian Revisi	Paraf
1.	Perbaiki flowchart.	
2.		

Palembang, Agustus 2022

Dosen Penguji,

(Emma Laila,S.Kom.,M.Kom.)

NIP. 197703292001121001

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 **Website :** www.polsri.ac.id **E-mail :** info@polsri.ac.id	
	REVISI TUGAS AKHIR (TA)	

Ruang	:	2
Dosen Penguji	:	Slamet Widodo,M.Kom.
Nama Mahasiswa	:	M. Willy Kurniawan
NIM	:	061930701649
Jurusan / Program Studi	:	Teknik Komputer / D3 Teknik Komputer
Judul LA / Skripsi	:	Rancang Bangun Alat Validasi Keanggotaan Perpustakaan Yang Memperhatikan Protokol Kesehatan Berbasis Mikrokontroler

No	Uraian Revisi	Paraf
	Revisi penyajian Bab IV Dijelaskan Hasil gambar	

Palembang, Agustus 2022
Dosen Penguji,

(Slamet Widodo,M.Kom.)
NIP. 197305162002121001

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI TUGAS AKHIR (TA)	

Ruang	:	2
Dosen Penguji	:	Isnainy Azro,S.Kom,M.Kom.
Nama Mahasiswa	:	M. Willy Kurniawan
NIM	:	061930701649
Jurusan / Program Studi	:	Teknik Komputer / D3 Teknik Komputer
Judul LA / Skripsi	:	Rancang Bangun Alat Validasi Keanggotaan Perpustakaan Yang Memperhatikan Protokol Kesehatan Berbasis Mikrokontroler

No	Uraian Revisi	Paraf

Palembang, Agustus 2022
Dosen Penguji,

(Isnainy Azro,S.Kom,M.Kom.)
NIP. 197310012002122002

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI TUGAS AKHIR (TA)	

Ruang	:	2
Dosen Penguji	:	Ikhthison Mekongga,S.T.,M.Kom.
Nama Mahasiswa	:	M. Willy Kurniawan
NIM	:	061930701649
Jurusan / Program Studi	:	Teknik Komputer / D3 Teknik Komputer
Judul LA / Skripsi	:	Rancang Bangun Alat Validasi Keanggotaan Perpustakaan Yang Memperhatikan Protokol Kesehatan Berbasis Mikrokontroler

No	Uraian Revisi	Paraf
	Format penulisan	

Palembang, Agustus 2022
Dosen Penguji,

(Ikhthison Mekongga,S.T.,M.Kom.)
NIP. 197705242000031002

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
	REVISI TUGAS AKHIR (TA)	

Ruang	:	2
Dosen Penguji	:	Adi Sutrisman,S.Kom.,M.Kom.
Nama Mahasiswa	:	M. Willy Kurniawan
NIM	:	061930701649
Jurusan / Program Studi	:	Teknik Komputer / D3 Teknik Komputer
Judul LA / Skripsi	:	Rancang Bangun Alat Validasi Keanggotaan Perpustakaan Yang Memperhatikan Protokol Kesehatan Berbasis Mikrokontroler

No	Uraian Revisi	Paraf
1	Lampirkan coding	

Palembang, 10 Agustus 2022
Dosen Penguji,

(Adi Sutrisman,S.Kom.,M.Kom.)
NIP. 197503052001121005

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JalanSrijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail :info@polsri.ac.id

PELAKSANAAN REVISI LAPORAN AKHIR (LA)

Mahasiswa berikut,

Nama	:	M. Willy Kurniawan
NIM	:	061930701649
Jurusan/Program Studi	:	Teknik Komputer/DIII-Teknik Komputer
Judul Laporan KP	:	Rancang Bangun Alat Validasi Keanggotaan Perpustakaan Yang Memperhatikan Protokol Kesehatan Berbasis Mikrokontroler

Telah melaksanakan revisi terhadap Laporan Akhir (LA) yang diseminarkan pada hari Kamis tanggal 28 bulan Juli tahun 2022 . Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penilai yang memberikan revisi:

No.	Komentar	Nama Dosen Penilai *)	Tanggal	Tanda Tangan
1	OK	Ema Laila,S.kom.,M.Kom.	22/8 22	
2	OK	Slamet Widodo,M.Kom.	25/8 2022	
3	OK	Isnainy Azro,M.Kom.	22/8 22	
4	OK	Ikhthison Mekongga,S.T.,M.Kom.	25/8 2022	
5	OK	Adi Sutrisman,M.Kom.	18/8 -22	

Palembang, ...

Ketua Penilai **)

Ema Laila,S.Kom.,M.Kom.

NIP. 197703292001122002