LAMPIRAN

Source Code

1. Program Arduino Nano

```
        //Memanggil Library//
//=================//
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
#include <SPI.h>
#include <MFRC522.h>
//Mendefinisikan LED Pada Pin arduino
//Lihat Gambar Wiring dan Samakan
#define ledR 6 // ledR = LED Red/Merah ke pin D6
#define ledG 7 // ledG = LED Green/Hijau ke pin D7
#define ledY 5 // ledy = LED Yellow/kuning ke pin D5
//Mendefinisikan Relay/Buzzer/dan erase Pada Pin arduino
//Lihat Gambar Wiring dan Samakan
#define relay 8 //relay Pada Digital 8
#define erase 3 //Pin erase D3 (konek ke Gnd)
int buzer=2;
boolean cocok = false;
boolean programMode = false;
int berhasil;
byte sementara[4];
byte baca[4];
byte master[4];
//Mendefinisikan Pin SS dan RST RFID ke pin Arduino
//Lihat Gambar Wiring dan Samakan
```

```
#define SS_PIN 10 //Pin SS ke Pin D10
#define RST_PIN 9 //Pin RST ke pin D9
MFRC522 mfrc522(SS_PIN, RST_PIN);
/*Menentukan Alamat i2c LCD dan Jenis LCD
 * Dimana Alamat I2C "0x27" (ganti Jika Alamatnya Berbeda)
 * Lihat Tutorial Scanner I2C
 * Untuk Jenis LCD yaitu 1602 atau 16x2
 */
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
  pinMode(buzer,OUTPUT); //Set Buzzer Sebagai Output dst.
  pinMode(ledR, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(ledY, OUTPUT);
  pinMode(relay, OUTPUT);
  pinMode(erase, INPUT_PULLUP);//Set erase Sebagai Input yang memakai Puuup internal
  //Seting Relay menjadi HIGH agar ketika Awal Program relay dalam keadaan Mati
  //Dimana Type Relay adalah Aktiv LOW (Low berarti AktiV dan sebaliknya)
  digitalWrite(relay, HIGH);
//Menyiapkan LCD dan Program tulisan LCD sebagai pembuakaan
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("HII..!!");
  lcd.setCursor(0,1);
  lcd.print("TAP MASTER CARD!");
  //Menyiapkan Komunikasi Serial untuk Menampilkan Pembacaan Real
  Serial.begin(9600);
  //Menyiapkan Komunikasi SPI karena RFID berkomunikasi menggunakan
SPI
  SPI.begin();
  //Menyiapakan Module RFID
  mfrc522.PCD_Init();
  ShowReaderDetails();
  //Program Dimana jika Tombol erase ditekan maka EEprom akan dihapus
  if (digitalRead(erase) == LOW)
  {
    digitalWrite(ledR, HIGH);
    digitalWrite(ledG, HIGH);
    delay(5000);
    if (digitalRead(erase) == LOW)
    {
      for (int x = 0; x < EEPROM.length(); x = x + 1)
      {
        if (EEPROM.read(x) == 0) {}
        else
        {
          EEPROM.write(x, 0);
        }
      }
      digitalWrite(buzer,HIGH);
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
      delay(200);
      digitalWrite(ledR, HIGH);
      digitalWrite(ledG, HIGH);
      delay(200);
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
      delay(200);
      digitalWrite(ledR, HIGH);
      digitalWrite(ledG, HIGH);
      delay(200);
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
    }
    else
    {
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
      digitalWrite(buzer,LOW);
    }
  }
void successWrite() //Program jika sukses menulis ke EEPROM
{
  digitalWrite(ledR,LOW);
  digitalWrite(ledG,LOW);
  digitalWrite(ledY,LOW);
  delay(200);
  digitalWrite(ledR, HIGH);
  digitalWrite(ledG, LOW);
  delay(200);
  digitalWrite(ledG, HIGH);
  digitalWrite(ledR, LOW);
  delay(200);
  digitalWrite(ledR, HIGH);
  digitalWrite(ledG, LOW);
  delay(200);
  digitalWrite(ledG, HIGH);
  digitalWrite(ledR, LOW);
  delay(200);
  digitalWrite(ledR, HIGH);
  digitalWrite(ledG, LOW);
  delay(200);
  digitalWrite(ledG, HIGH);
  digitalWrite(ledR, LOW);
  delay(200);
  digitalWrite(ledG, LOW);
}
boolean findID( byte find[] )
{
  int count = EEPROM.read(0);
  for ( int i = 1; i <= count; i++ )
  {
    readID(i);
    if ( checkTwo( find, sementara ) )
    {
      return true;
      break;
    }
    else
    {
    }
  }
  return false;
}
void failedWrite() //Program jika GAGAL menulis ke EEPROM
{
  digitalWrite(ledR, HIGH);
      digitalWrite(ledG, HIGH);
      delay(200);
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
      delay(200);
      digitalWrite(ledR, HIGH);
      digitalWrite(ledG, HIGH);
      delay(200);
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
}
void deleteID( byte a[] ) //Program menghapus ID pada EEPROM
{
  if ( !findID( a ) )
  {
    failedWrite();
```

```
 // Serial.println(F("Failed! There is something wrong with ID or bad DATABASE"));
 }
 else
 {
  int num = EEPROM.read(0);
  int slot;
  int start;
  int looping;
  int j;
  int count = EEPROM.read(0);
  slot = findIDSLOT( a );
  start = (slot * 4) + 2;
  looping = ((num - slot) * 4);
  num--;
  EEPROM.write( 0, num );
  for ( j = 0; j < looping; j++ )
  {
   EEPROM.write( start + j, EEPROM.read(start + 4 + j));
  }
  for ( int k = 0; k < 4; k++ ) {
   EEPROM.write( start + j + k, 0);
  }
  successDelete();
    digitalWrite(buzer,HIGH);
    delay(1000);
    digitalWrite(buzer,LOW);
    delay(1000);
    digitalWrite(buzer,HIGH);
    delay(1000);
    digitalWrite(buzer,LOW);
    delay(1000);
    digitalWrite(buzer,HIGH);
    delay(1000);
    digitalWrite(buzer,LOW);
    // Serial.println(F("Succesfully removed ID record from DATABASE"));
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Sukses Tererase");
    lcd.setCursor(0,1);
    lcd.print("Tap Mstr To Cfrm!");
 }
}
boolean checkTwo ( byte a[], byte b[] )
{
 if ( a[0] != NULL )
  cocok = true;
 for ( int k = 0; k < 4; k++ )
 {
  if ( a[k] != b[k] )
   cocok = false;
 }
 if ( cocok )
 {
  return true;
 }
```

```
  else
  {
    return false;
  }
}
void readID( int number )
{
  int start = (number * 4 ) + 2;
  for ( int i = 0; i < 4; i++ )
  {
    sementara[i] = EEPROM.read(start + i);
  }
}
int findIDSLOT( byte find[] )
{
  int count = EEPROM.read(0);
  for ( int i = 1; i <= count; i++ )
  {
    readID(i);
    if ( checkTwo( find, sementara ) )
    {
      return i;
      break;
    }
  }
}
void successDelete() //Program jika Penghapusan berhasil
{
  digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
      delay(200);
      digitalWrite(ledR, HIGH);
      digitalWrite(ledG, HIGH);
      delay(200);
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
      delay(200);
      digitalWrite(ledR, HIGH);
      digitalWrite(ledG, HIGH);
      delay(200);
      digitalWrite(ledR, LOW);
      digitalWrite(ledG, LOW);
}
void ShowReaderDetails()
{
  byte v = mfrc522.PCD_ReadRegister(mfrc522.VersionReg);
  // Serial.print(F("MFRC522 Version: 0x"));
  // Serial.print(v, HEX);
  if (v == 0x91)
  {}
  else if (v == 0x11)
  {}
  else{}
  if ((v == 0x00) || (v == 0xFF))
  {
    while(true);
  }
}
```

2. Program ESP32

```
        #define BLYNK_TEMPLATE_ID "TMPL6hma7o9Vv"
#define BLYNK_TEMPLATE_NAME "isti"
#define                                    BLYNK_AUTH_TOKEN
"PkTHrZUShRzzhJllHU9WozlTlvGCEXVP"
#define BLYNK_PRINT Serial //Kita menggunakan Blynk serial
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <HardwareSerial.h>
// Inisialisasi Serial Port kedua
HardwareSerial SerialPort(2);
// Token Auth Blynk
char auth[] = BLYNK_AUTH_TOKEN;
// Detail WiFi
char ssid[] = "istyyy";
char pass[] = "istirahma";
// Variabel untuk menyimpan pesan
String receivedMessage;
void setup() {
  Serial.begin(9600); // Inisialisasi Serial monitor
  SerialPort.begin(9600, SERIAL_8N1, 16, 17); // Inisialisasi SerialPort kedua
  // Koneksi ke WiFi
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnected to WiFi");
  // Inisialisasi Blynk
  Blynk.begin(auth, ssid, pass);
}
void loop() {
  // Blynk loop
  Blynk.run();
  // Cek apakah ada data yang diterima
  while (SerialPort.available() > 0) {
    char receivedChar = SerialPort.read();
    if (isPrintable(receivedChar) || receivedChar == '\n') {   // Hanya terima
karakter yang dapat dicetak dan newline
      if (receivedChar == '\n') {
        if (receivedMessage == "penyusup") {
          Serial.println(receivedMessage);   // Print the received message in the
Serial monitor
          Blynk.logEvent("penyusup"); // Kirim log event ke Blynk
        } else if (receivedMessage == "terbuka") {
          Serial.println(receivedMessage);   // Print the received message in the
Serial monitor
          Blynk.logEvent("terbuka"); // Kirim log event ke Blynk
        }
        receivedMessage = "";  // Reset the received message
      } else {
        receivedMessage += receivedChar;  // Append characters to the received
message
      }
```