

## LAMPIRAN KODE PROGRAM RFID *MULTI-READER* MENGUNAKAN ALGORITMA *DISTRIBUTED COLOR* *SELECTION*

```
#include <SPI.h>           // include library SPI
#include <MFRC522.h>       // include library RFID MFRC522

//configurasi PIN
#define RST0_PIN          9
#define SS_0_PIN          10
#define RST1_PIN          6
#define SS_1_PIN          8

#define jumlahreader      2 //init jumlah reader yg digunakan

byte sspin[] = {SS_0_PIN,SS_1_PIN}; //init pin ss
byte rstpin[] = {RST0_PIN, RST1_PIN}; //init pin rst
int slot0;
int slot1;           //color/slot tiap reader

MFRC522 rfid[jumlahreader]; //Create MFRC522 instance

void setup() {
  Serial.begin(9600); //init komunikasi serial dengan PC
  SPI.begin(); //init SPI bus

  for (int reader = 0; reader < jumlahreader; reader++) {
    rfid[reader].PCD_Init(sspin[reader], rstpin[reader]); //init
    tiap RFID reader
    rfid[reader].PCD_SetAntennaGain(rfid[reader].RxGain_max); //
    mengoptimalkan jangkauan reader (opsional)
    // Cek Reader
    Serial.print("Reader ");
    Serial.print(reader);
    Serial.print(": ");
    rfid[reader].PCD_DumpVersionToSerial(); // cek firmware(opsional)
  }
  randomSeed(analogRead(A0));
  slot0 = random(2); //memilih nomor slot secara acak
  slot1 = random(2); //memilih nomor sot secara acak
```

```

while(slot0 == slot1){
    Serial.println("Selection Color Fail, Re-Pick.....");
    slot0 = random(2); //memilih nomor slot secara acak
    slot1 = random(2); //memilih nomor sot secara acak
}
Serial.println("Selection Color Status = OK");
Serial.print("Reader 0 Choose Color : ");
Serial.println(slot0);
Serial.print("Reader 1 Choose Color : ");
Serial.println(slot1);

    Serial.println("All Reader Ready");
    Serial.println("Scanning. . . . .");
}

void loop() {
    unsigned long m = millis(); // inisialisasi waktu millisecond
    // setting awal RFID mati
    if (m < 1000) {
        for (int reader = 0; reader < jumlahreader; reader++) {
            digitalWrite(rstpin[reader], LOW); //Fungsi reset dan mematikan
RFID
            digitalWrite(sspin[reader], LOW); //Fungsi chip select, memilih
RFID yg dimatikan
        }
    }
    // mulai masuk pembacaan
    for (int reader = 0; reader < jumlahreader; reader++) {
//menghitung jumlah reader
        //=====
        // mulai reader 0
        //=====
        //pembacaan slot ke-1
        if (reader == 0 && slot0 == 0) { // jika reader ke-0 dan color yg
dipilih = 1
            if (m >= 1100 && m < 5000 ) { // pada waktu sekian baca tag
                rfid[reader].PCD_Init(sspin[reader], rstpin[reader]); //pin
harus diinisialisasi ulang pada tiap slot, karena klau hanya diawal
maka hanya akan terbaca 1 reader saja
                rfid[reader].PCD_SetAntennaGain(rfid[reader].RxGain_max);
//extend antena
                //mulai baca
                if (rfid[reader].PICC_IsNewCardPresent() &&
rfid[reader].PICC_ReadCardSerial()) {

```

```

        Serial.print("Reader ");
        Serial.print(reader);
        Serial.print(": UID: ");
        uid(rfid[reader].uid.uidByte, rfid[reader].uid.size);
        Serial.println();

    }
    else {
        //stop baca
        digitalWrite(rstpin[reader], LOW);
        digitalWrite(sspin[reader], LOW);
    }
}
//pembacaan slot ke-2
else if (reader == 0 && slot0 == 1) {
    if (m >= 5000 && m < 10000 ) {
        rfid[reader].PCD_Init(sspin[reader], rstpin[reader]); //pin
        harus diinisialisasi ulang pada tiap slot, karena klau hanya diawal
        maka hanya akan terbaca 1 reader saja
        rfid[reader].PCD_SetAntennaGain(rfid[reader].RxGain_max);
        //extend antena
        //mulai baca
        if (rfid[reader].PICC_IsNewCardPresent() &&
rfid[reader].PICC_ReadCardSerial()) {
            Serial.print("Reader ");
            Serial.print(reader);
            Serial.print(": UID: ");
            uid(rfid[reader].uid.uidByte, rfid[reader].uid.size);
            Serial.println();

        }
        else {
            //stop baca
            digitalWrite(rstpin[reader], LOW);
            digitalWrite(sspin[reader], LOW);
        }
    }
}

//=====
// mulai reader 1

```

```

//=====
//pembacaan slot ke-1
//-----
else if (reader == 1 && slot1 == 0) {
    if (m >= 1000 && m < 5000 ) {
        rfid[reader].PCD_Init(sspin[reader], rstpin[reader]); //pin
        harus diinisialisasi ulang pada tiap slot, karena klau hanya diawal
        maka hanya akan terbaca 1 reader saja
        rfid[reader].PCD_SetAntennaGain(rfid[reader].RxGain_max);
//extend antena
        //mulai baca
        if (rfid[reader].PICC_IsNewCardPresent() &&
rfid[reader].PICC_ReadCardSerial()) {
            Serial.print("Reader ");
            Serial.print(reader);
            Serial.print(": UID: ");
            uid(rfid[reader].uid.uidByte, rfid[reader].uid.size);
            Serial.println();
        }
        else {
            //stop baca
            digitalWrite(rstpin[reader], LOW);
            digitalWrite(sspin[reader], LOW);
        }
    }
}
//pembacaan slot ke-2
//-----
else if (reader == 1 && slot1 == 1) {
    if (m >= 5100 && m < 10000 ) {
        rfid[reader].PCD_Init(sspin[reader], rstpin[reader]); //pin
        harus diinisialisasi ulang pada tiap slot, karena klau hanya diawal
        maka hanya akan terbaca 1 reader saja
        rfid[reader].PCD_SetAntennaGain(rfid[reader].RxGain_max);
//extend antena
        //mulai baca
        if (rfid[reader].PICC_IsNewCardPresent() &&
rfid[reader].PICC_ReadCardSerial()) {
            Serial.print("Reader ");
            Serial.print(reader);
            Serial.print(": UID: ");
            uid(rfid[reader].uid.uidByte, rfid[reader].uid.size);
            Serial.println();
        }
    }
}

```

```
    else {
        //stop baca
        digitalWrite(rstpin[reader], LOW);
        digitalWrite(sspin[reader], LOW);
    }
}
}
}

void uid(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);
    }
}
```