Koding Sistem Penataan Lahan Parkir Menggunakan Arduino

```
/* Include the standard Arduino SPI library */
#include <SPI.h>
/* Include the RFID library */
#include <RFID.h>

/* Define the DIO used for the SDA (SS) and RST (reset) pins. */
#define SDA_DIO 9
#define RESET_DIO 8
/* Create an instance of the RFID library */
RFID RC522(SDA_DIO, RESET_DIO);

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

Servo myservo2;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int trig = 26;          // membuat varibel trig yang di set ke-pin 3
int echo = 27;          // membuat variabel echo yang di set ke-pin 2
long durasi, jarak;     // membuat variabel durasi dan jarak

int trig2 = 48;          // membuat varibel trig yang di set ke-pin 3
int echo2 = 46;          // membuat variabel echo yang di set ke-pin 2
long durasi2, jarak2;     // membuat variabel durasi dan jarak

int trig3 = 36;          // membuat varibel trig yang di set ke-pin 3
int echo3 = 37;          // membuat variabel echo yang di set ke-pin 2
long durasi3, jarak3;     // membuat variabel durasi dan jarak

String inputString = "";         // a String to hold incoming data
boolean stringComplete = false;  // whether the string is complete

#define parkir1 A0
#define parkir2 A1
#define parkir3 A2
#define parkir4 A3
#define parkir5 A4
#define parkir6 A5
#define parkir7 A6
#define parkir8 A7
#define parkir9 A8
#define parkir10 A9
```

```
#define parkir11 A10
#define parkir12 A11

#define parkir6 A5
int nParkir1 = 0;
int nParkir2 = 0;
int nParkir3 = 0;
int nParkir4 = 0;
int nParkir5 = 0;
int nParkir6 = 0;
int nParkir7 = 0;
int nParkir8 = 0;
int nParkir9 = 0;
int nParkir10 = 0;
int nParkir11 = 0;
int nParkir12 = 0;

bool sparkir1 = 0;
bool sparkir2 = 0;
bool sparkir3 = 0;
bool sparkir4 = 0;
bool sparkir5 = 0;
bool sparkir6 = 0;
bool sparkir7 = 0;
bool sparkir8 = 0;
bool sparkir9 = 0;
bool sparkir10 = 0;
bool sparkir11 = 0;
bool sparkir12 = 0;

bool seharusnya_parkir1 = 1;
bool seharusnya_parkir2 = 1;
bool seharusnya_parkir3 = 1;
bool seharusnya_parkir4 = 1;
bool seharusnya_parkir5 = 1;
bool seharusnya_parkir6 = 1;
bool seharusnya_parkir7 = 1;
bool seharusnya_parkir8 = 1;
bool seharusnya_parkir9 = 1;
bool seharusnya_parkir10 = 1;
bool seharusnya_parkir11 = 1;
bool seharusnya_parkir12 = 1;

#define buzzer1 30
#define buzzer2 34

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

```
  /* Enable the SPI interface */
  SPI.begin();
  /* Initialise the RFID reader */
  RC522.init();
  myservo.attach(2);
  tutup_servo1();
  pinMode(trig, OUTPUT);    // set pin trig menjadi OUTPUT
  pinMode(echo, INPUT);     // set pin echo menjadi INPUT

  myservo2.attach(4);
  tutup_servo2();
  pinMode(trig2, OUTPUT);    // set pin trig menjadi OUTPUT
  pinMode(echo2, INPUT);     // set pin echo menjadi INPUT

  buka_servo1();
  buka_servo2();

  pinMode(parkir1, INPUT);
  pinMode(parkir2, INPUT);
  pinMode(parkir3, INPUT);
  pinMode(parkir4, INPUT);
  pinMode(parkir5, INPUT);
  pinMode(parkir6, INPUT);
  pinMode(parkir7, INPUT);
  pinMode(parkir8, INPUT);
  pinMode(parkir9, INPUT);
  pinMode(parkir10, INPUT);
  pinMode(parkir11, INPUT);
  pinMode(parkir12, INPUT);

  pinMode (buzzer1, OUTPUT);
  digitalWrite (buzzer1, LOW);

  pinMode (buzzer2, OUTPUT);
  digitalWrite (buzzer2, LOW);
}

void loop() {

  baca_sensor ();
  sensor_ultrasonik3 ();

  Serial.print(sparkir1);
  Serial.print(" ");
  Serial.print(sparkir2);
  Serial.print(" ");
  Serial.print(sparkir3);
  Serial.print(" ");
```

```
Serial.print(sparkir4);
Serial.print(" ");
Serial.print(sparkir5);
Serial.print(" ");
Serial.print(sparkir6);
Serial.print(" ");
Serial.print(sparkir7);
Serial.print(" ");
Serial.print(sparkir8);
Serial.print(" ");
Serial.print(sparkir9);
Serial.print(" ");
Serial.print(sparkir10);
Serial.print(" ");
Serial.print(sparkir11);
Serial.print(" ");
Serial.print(sparkir12);
Serial.println(" ");

Serial.print(seharusnya_parkir1 );
Serial.print(" ");
Serial.print(seharusnya_parkir2 );
Serial.print(" ");
Serial.print(seharusnya_parkir3 );
Serial.print(" ");
Serial.print(seharusnya_parkir4 );
Serial.print(" ");
Serial.print(seharusnya_parkir5 );
Serial.print(" ");
Serial.print(seharusnya_parkir6 );
Serial.print(" ");
Serial.print(seharusnya_parkir7 );
Serial.print(" ");
Serial.print(seharusnya_parkir8 );
Serial.print(" ");
Serial.print(seharusnya_parkir9 );
Serial.print(" ");
Serial.print(seharusnya_parkir10 );
Serial.print(" ");
Serial.print(seharusnya_parkir11 );
Serial.print(" ");
Serial.print(seharusnya_parkir12 );
Serial.println(" ");

if (seharusnya_parkir1 == 0 && sparkir1 == 1) {
  hidup_buzzer1();
} else if (seharusnya_parkir2 == 0 && sparkir2 == 1) {
  hidup_buzzer1 ();
```

```
} else if (seharusnya_parkir3 == 0 && sparkir3 == 1) {
  hidup_buzzer1 ();
} else if (seharusnya_parkir4 == 0 && sparkir4 == 1) {
  hidup_buzzer1 ();
} else if (seharusnya_parkir5 == 0 && sparkir5 == 1) {
  hidup_buzzer1 ();
} else if (seharusnya_parkir6 == 0 && sparkir6 == 1) {
  hidup_buzzer1 ();
} else {
  mati_buzzer1 ();
}

if (seharusnya_parkir7 == 0 && sparkir7 == 1) {
  hidup_buzzer2();
} else if (seharusnya_parkir8 == 0 && sparkir8 == 1) {
  hidup_buzzer2 ();
} else if (seharusnya_parkir9 == 0 && sparkir9 == 1) {
  hidup_buzzer2 ();
} else if (seharusnya_parkir10 == 0 && sparkir10 == 1) {
  hidup_buzzer2 ();
} else if (seharusnya_parkir11 == 0 && sparkir11 == 1) {
  hidup_buzzer2 ();
} else if (seharusnya_parkir12 == 0 && sparkir12 == 1) {
  hidup_buzzer2 ();
} else {
  mati_buzzer2 ();
}

serialEvent() ;
// print the string when a newline arrives:
if (stringComplete) {
  if (inputString.indexOf("Masuk") != -1) {
    buka_servo1();
  } else if (inputString.indexOf("Keluar") != -1) {
    buka_servo2();
  } else if (inputString.indexOf("Kondisi") != -1) {
    if (getValue(inputString, ',', 1).indexOf("Ada") != -1) {
      seharusnya_parkir1 = 1;
    } else {
      seharusnya_parkir1 = 0;
    }

    if (getValue(inputString, ',', 2).indexOf("Ada") != -1) {
      seharusnya_parkir2 = 1;
    } else {
      seharusnya_parkir2 = 0;
    }
```

```
if (getValue(inputString, ',', 3).indexOf("Ada") != -1) {
  seharusnya_parkir3 = 1;
} else {
  seharusnya_parkir3 = 0;
}

if (getValue(inputString, ',', 4).indexOf("Ada") != -1) {
  seharusnya_parkir4 = 1;
} else {
  seharusnya_parkir4 = 0;
}

if (getValue(inputString, ',', 5).indexOf("Ada") != -1) {
  seharusnya_parkir5 = 1;
} else {
  seharusnya_parkir5 = 0;
}

if (getValue(inputString, ',', 6).indexOf("Ada") != -1) {
  seharusnya_parkir6 = 1;
} else {
  seharusnya_parkir6 = 0;
}
if (getValue(inputString, ',', 7).indexOf("Ada") != -1) {
  seharusnya_parkir7 = 1;
} else {
  seharusnya_parkir7 = 0;
}
if (getValue(inputString, ',', 8).indexOf("Ada") != -1) {
  seharusnya_parkir8 = 1;
} else {
  seharusnya_parkir8 = 0;
}
if (getValue(inputString, ',', 9).indexOf("Ada") != -1) {
  seharusnya_parkir9 = 1;
} else {
  seharusnya_parkir9 = 0;
}
if (getValue(inputString, ',', 10).indexOf("Ada") != -1) {
  seharusnya_parkir10 = 1;
} else {
  seharusnya_parkir10 = 0;
}
if (getValue(inputString, ',', 11).indexOf("Ada") != -1) {
  seharusnya_parkir11 = 1;
} else {
  seharusnya_parkir11 = 0;
}
```

```
      if (getValue(inputString, ',', 12).indexOf("Ada") != -1) {
        seharusnya_parkir12 = 1;
      } else {
        seharusnya_parkir12 = 0;
      }
    }
    Serial.println(inputString);
    // clear the string:
    inputString = "";
    stringComplete = false;
  }
  /* Has a card been detected? */
  sensor_ultrasonik ();
  //  delay(1000);
  sensor_ultrasonik2 ();
  //  delay(1000);
  if (jarak < 10 && jarak!=0) {
    tutup_servo1();
  }
  if (jarak2 < 10) {
    tutup_servo2();
  }
  if (RC522.isCard())
  {
    /* If so then get its serial number */
    RC522.readCardSerial();
    String data_rfid = "";
    for (int i = 0; i < 5; i++)
    {
      //data_rfid.setCharAt(i,RC522.serNum[i]);
      data_rfid = data_rfid + RC522.serNum[i];
      //Serial.print(RC522.serNum[i]);
      //Serial.print(RC522.serNum[i],HEX); //to print card detail in Hexa Decimal
format
    }
    Serial.println(data_rfid);
    data_rfid = enkripsi(data_rfid);
    Serial.print("in,");
    Serial.println(data_rfid);
//    Serial.println();
    delay(5000);
    //    Serial.println();
  }

}

void buka_servo1() {
  myservo.write(90);
```

```
  delay(500);
}
void tutup_servo1() {
  myservo.write(165);
}
void buka_servo2() {
  myservo2.write(120);
  delay(500);
}
void tutup_servo2() {
  myservo2.write(40);
}
void sensor_ultrasonik () {
  // program dibawah ini agar trigger memancarakan suara ultrasonic
  digitalWrite(trig, LOW);
  delayMicroseconds(20);
  digitalWrite(trig, HIGH);
  delayMicroseconds(20);
  digitalWrite(trig, LOW);
  //  delayMicroseconds(8);

  durasi = pulseIn(echo, HIGH); // menerima suara ultrasonic
  //  durasi = 0;
  //  while (digitalRead(echo)==1){
  ////    durasi++;
  //  }
  //  while (digitalRead(echo)==0){
  //    durasi++;
  //  }
  jarak = (durasi / 2) / 29.1;  // mengubah durasi menjadi jarak (cm)
  Serial.println(jarak);        // menampilkan jarak pada Serial Monitor
}

void sensor_ultrasonik2 () {
  // program dibawah ini agar trigger memancarakan suara ultrasonic
  digitalWrite(trig2, LOW);
  delayMicroseconds(20);
  digitalWrite(trig2, HIGH);
  delayMicroseconds(20);
  digitalWrite(trig2, LOW);
  //  delayMicroseconds(8);

  durasi2 = pulseIn(echo2, HIGH); // menerima suara ultrasonic
  //  durasi2 = 0;
  //  while (digitalRead(echo2)==1){
  ////    durasi2++;
  //  }
  //    while (digitalRead(echo2)==0){
```

```
  //    durasi2++;
  //  }
  jarak2 = (durasi2 / 2) / 29.1;  // mengubah durasi menjadi jarak (cm)
  Serial.println(jarak2);        // menampilkan jarak pada Serial Monitor
}

void sensor_ultrasonik3 () {
  // program dibawah ini agar trigger memancarakan suara ultrasonic
  digitalWrite(trig3, LOW);
  delayMicroseconds(20);
  digitalWrite(trig3, HIGH);
  delayMicroseconds(20);
  digitalWrite(trig3, LOW);
  //  delayMicroseconds(8);

  durasi3 = pulseIn(echo3, HIGH); // menerima suara ultrasonic
  //  durasi2 = 0;
  //  while (digitalRead(echo2)==1){
  ////    durasi2++;
  //  }
  //    while (digitalRead(echo2)==0){
  //    durasi2++;
  //  }
  jarak3 = (durasi3 / 2) / 29.1;  // mengubah durasi menjadi jarak (cm)
  Serial.println(jarak3);        // menampilkan jarak pada Serial Monitor
  if (jarak3 < 8) {
    Serial.println("silahkan_masuk");
  }
}

void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    // if the incoming character is a newline, set a flag so the main loop can
    // do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    } else if (inChar == '\r') {

    } else {
      inputString += inChar;
    }
  }
}

void baca_sensor () {
```

```
nParkir1 = analogRead(parkir1);
//  Serial.print(nParkir1);
//  Serial.print(" Parkir 1 = ");
sparkir1 = apakah_ada_mobil(nParkir1, 800);
//  Serial.println(apakah_ada_mobil(nParkir1,950));

nParkir2 = analogRead(parkir2);
//Serial.print(nParkir2);
//Serial.print(" Parkir 2 = ");
sparkir2 = apakah_ada_mobil(nParkir2, 850);
//Serial.println(apakah_ada_mobil(nParkir2,950));


nParkir3 = analogRead(parkir3);
//Serial.print(nParkir3);
//Serial.print(" Parkir 3 = ");
sparkir3 = apakah_ada_mobil(nParkir3, 850);
//Serial.println(apakah_ada_mobil(nParkir3,950));

nParkir4 = analogRead(parkir4);
//Serial.print(nParkir4);
//Serial.print(" Parkir 4 = ");
sparkir4 = apakah_ada_mobil(nParkir4, 750);
//Serial.println(apakah_ada_mobil(nParkir4,930));

nParkir5 = analogRead(parkir5);
//Serial.print(nParkir5);
//Serial.print(" Parkir 5 = ");
sparkir5 = apakah_ada_mobil(nParkir5, 820);
//Serial.println(apakah_ada_mobil(nParkir5,950));

nParkir6 = analogRead(parkir6);
//Serial.print(nParkir6);
//Serial.print(" Parkir 6 = ");
sparkir6 = apakah_ada_mobil(nParkir6, 750);
//Serial.println(apakah_ada_mobil(nParkir6,950));

nParkir7 = analogRead(parkir7);
//Serial.print(nParkir6);
//Serial.print(" Parkir 6 = ");
sparkir7 = apakah_ada_mobil(nParkir7, 750);
//Serial.println(apakah_ada_mobil(nParkir6,950));

nParkir8 = analogRead(parkir8);
//Serial.print(nParkir6);
//Serial.print(" Parkir 6 = ");
sparkir8 = apakah_ada_mobil(nParkir8, 750);
//Serial.println(apakah_ada_mobil(nParkir6,950));
```

```
nParkir9 = analogRead(parkir9);
//Serial.print(nParkir6);
//Serial.print(" Parkir 6 = ");
sparkir9 = apakah_ada_mobil(nParkir9, 750);
//Serial.println(apakah_ada_mobil(nParkir6,950));

nParkir10 = analogRead(parkir10);
//Serial.print(nParkir6);
//Serial.print(" Parkir 6 = ");
sparkir10 = apakah_ada_mobil(nParkir10, 750);
//Serial.println(apakah_ada_mobil(nParkir6,950));

nParkir11 = analogRead(parkir11);
//Serial.print(nParkir6);
//Serial.print(" Parkir 6 = ");
sparkir11 = apakah_ada_mobil(nParkir11, 759);
//Serial.println(apakah_ada_mobil(nParkir6,950));

nParkir12 = analogRead(parkir12);
//Serial.print(nParkir6);
//Serial.print(" Parkir 6 = ");
sparkir12 = apakah_ada_mobil(nParkir12, 750);
//Serial.println(apakah_ada_mobil(nParkir6,950));

Serial.print("parkir,");
Serial.print(sparkir1);
Serial.print(",");
Serial.print(sparkir2);
Serial.print(",");
Serial.print(sparkir3);
Serial.print(",");
Serial.print(sparkir4);
Serial.print(",");
Serial.print(sparkir5);
Serial.print(",");
Serial.print(sparkir6);
Serial.print(",");
Serial.print(sparkir7);
Serial.print(",");
Serial.print(sparkir8);
Serial.print(",");
Serial.print(sparkir9);
Serial.print(",");
Serial.print(sparkir10);
Serial.print(",");
Serial.print(sparkir11);
Serial.print(",");
```

```
  Serial.println(sparkir12);

}
bool apakah_ada_mobil(int nilai, int batas) {
  if (nilai > batas) {
    return 0;
  } else {
    return 1;
  }
}

void hidup_buzzer1 () {
  digitalWrite (buzzer1, HIGH);
}
void mati_buzzer1 () {
  digitalWrite (buzzer1, LOW);
}

void hidup_buzzer2 () {
  digitalWrite (buzzer2, HIGH);
}
void mati_buzzer2 () {
  digitalWrite (buzzer2, LOW);
}

String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = { 0, -1 };
  int maxIndex = data.length() - 1;

  for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
      found++;
      strIndex[0] = strIndex[1] + 1;
      strIndex[1] = (i == maxIndex) ? i + 1 : i;
    }
  }
  return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

String enkripsi(String plaintext)
{
  int kunci[2][2];
  String hasil_enkrip = "";
  kunci[0][0] = 3;
  kunci[0][1] = 3;
  kunci[1][0] = 2;
```

```
  kunci[1][1] = 5;

  //plaintext dipecah jadi array 2 2 2
  //cek dulu genap atau ganjil
  if (plaintext.length() % 2 == 0)
  {
    //genap
  }
  else
  {
    plaintext += " ";
  }
  //jadikan char array dahulu
//  char plaintext_array[] = plaintext.ToCharArray();

  //jadikan 2 2
  for (int i = 0; i < plaintext.length(); i += 2)
  {
    char c1 = plaintext.charAt(i);
    char c2 = plaintext.charAt(i+1);

    int n1 = kunci[0][0] * c1;
    int n2 = kunci[0][1] * c2;
    int n3 = kunci[1][0] * c1;
    int n4 = kunci[1][1] * c2;
    int n5 = n1 + n2;
    int n6 = n3 + n4;
    int n7 = n5 % 256;
    int n8 = n6 % 256;

    char e1 = (char)n7;
    char e2 = (char)n8;
//    Serial.print(e1);
//    Serial.print(" ");
//    Serial.println(e2);

    hasil_enkrip = hasil_enkrip +  e1;
    hasil_enkrip = hasil_enkrip +  e2;

//    hasil_enkrip.setCharAt(i,e1);
//    hasil_enkrip.setCharAt(i+1,e2);

//    Serial.println(hasil_enkrip);

  }
  //var test_baca = hasil_enkrip.ToCharArray();
  return hasil_enkrip;
}
```