Lampiran A

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST "skripsiektp-d807a.firebaseio.com"
#define FIREBASE_AUTH "Rc5bG1Ufu8tbh3ywlpPpKNRG7kzKc2C6vXPRXpYs"
#define WIFI_SSID "LG K10"
#define WIFI_PASSWORD "aribawonoirin"
//#define WIFI_SSID "gap"
//#define WIFI_PASSWORD "02011997g"

SoftwareSerial mySerial(D2, D3 ); // RX, TX
uint8_t value[7] = {0,0,0,0,0,0,0};

unsigned long previousMillisServo = 0;
unsigned long previousMillisDisplay = 0;
int sudut = 0;
int slot = 2;
int isCarDetected = 0;
bool tag = true;
bool tagDelay = false;
bool firstClose = true;
int slotAll = 0;
int temp = 0;
int used = 0;
int mode = 1;
int nonktp = 0;

String status = "";
```

```
Servo servo;

void setup() {
  // Open serial communications and wait for port to open:

  Serial.begin(4800);
  mySerial.begin(4800);
  pinMode(D2, INPUT);
  pinMode(D3, OUTPUT);
  pinMode(D7, INPUT);
  servo.attach(D6); //D4

  servo.write(0);
  delay(2000);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);


}

void loop() { // run over and over
  mode = Firebase.getInt("mode_out");
  slot = Firebase.getInt("slot");
  slotAll = Firebase.getInt("slotAll");
```

```
nonktp = Firebase.getInt("nonktp");
//
// Mobil terdeteksi di portal dan menunggu scan kartu
//
switch (mode) {
  case 1 :
    if (digitalRead(D7)==0){
      while (mySerial.available()>0) {
        mySerial.readBytes(value,7) ;
        //
        // Validasi ID sesuai database
        //
        if
    ((value[0]+value[1]+value[2]+value[3]+value[4]+value[5]+value[6
    ])>0){
                temp                                                =
    value[0]+value[1]+value[2]+value[3]+value[4]+value[5]+value[6];
              }
  //
  //      for (int i=0; i<7;i++){
  //        Serial.print(value[i],HEX);
  //        Serial.print(" ");
  //      }
  //      Serial.println();

        if (temp>0){
          status                                                    =
    Firebase.getString("user/"+String(value[0]+value[1]+value[2]+va
    lue[3]+value[4]+value[5]+value[6])+"/status");

          //Serial.println(status);
          if (!Firebase.failed()){
            if (status == "parkir"){
              //Serial.println("Mau buka");
              openGate(); // Buka gerbang
```

```
          firstClose = true;

Firebase.setString("user/"+String(value[0]+value[1]+value[2]+va
lue[3]+value[4]+value[5]+value[6])+"/status", "tidakparkir");
          if (tag){
            tag =false;
            tagDelay = true;
          }
        }
      } else {
        value[0] = 0x00; value[1] = 0x00; value[2] = 0x00;
        value[3] = 0x00; value[4] = 0x00; value[5] = 0x00;
        value[6] = 0x00;
      }
    }
  }
} else{
  //
  // Untuk delay sebelum portal tertutup
  //
  if (tagDelay){
    previousMillisServo = millis();
    tagDelay = false;
    slot++;
    used = slotAll - slot;
    Firebase.setInt("slot",slot);
    Firebase.setInt("used",used);
  }
  if (firstClose) {
    firstClose = false;
    if (temp>0){
      String                          nama                          =
Firebase.getString("user/"+String(temp)+"/nama");
      Firebase.remove("parking/"+String(temp));
    }
```

```
    }
    closeGate();
    tag = true;
  }
  //Serial.println(digitalRead(D4));
break;
case 2 :
  //
  // Input Manual Plat Nomor
  //

  String plat = Firebase.getString("inputmanual_out");
  String platData = Firebase.getString("parking/"+plat);
  if (digitalRead(D7)==0  && platData=="Anonymous"){
    openGate();
    tagDelay = true;
    firstClose=true;
  } else{
      //
      // Untuk delay sebelum portal tertutup
      //
      if (tagDelay){
        previousMillisServo = millis();
        tagDelay = false;
        slot++;
        used = slotAll - slot;
        nonktp--;
        Firebase.setInt("slot",slot);
        Firebase.setInt("used",used);
      }
      if (firstClose) {
        firstClose = false;
        String plat = Firebase.getString("inputmanual_out");
        Firebase.remove("parking/"+plat);
        Firebase.setString("inputmanual_out","");
```

```
            Firebase.setInt("mode_out",1);
            Firebase.setInt("nonktp",nonktp);
          }
          closeGate();

          tag = true;
        }
      break;
  }
}

//void showSlotPark(int slot){
//  if((millis() - previousMillisDisplay >= 1000))
//  {
//    previousMillisDisplay = millis();
//  }
//  display.showNumberDec(slot, false);
//}

void openGate(){
  if(millis() - previousMillisServo >= 2000)
  {
    previousMillisServo = millis();
    servo.write(90);
    //Serial.println("Udah buka");
  }
}

void closeGate(){
  if(millis() - previousMillisServo >= 2000)
  {
    previousMillisServo = millis();
    servo.write(0);
    //Serial.println("Udah tutup");
  }
```

Lampiran B

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include <Arduino.h>
#include <TM1637Display.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST "skripsiektp-d807a.firebaseio.com"
#define FIREBASE_AUTH "Rc5bG1Ufu8tbh3ywlpPpKNRG7kzKc2C6vXPRXpYs"
#define WIFI_SSID "LG K10"
#define WIFI_PASSWORD "aribawonoirin"
//#define WIFI_SSID "tugasakhir"
//#define WIFI_PASSWORD "12345678"
//#define WIFI_SSID "gap"
//#define WIFI_PASSWORD "02011997g"


// Module connection pins (Digital Pins)
#define CLK D1
#define DIO D0

SoftwareSerial mySerial(D2, D3 ); // RX, TX
uint8_t value[7] = {0,0,0,0,0,0,0};

unsigned long previousMillisServo = 0;
unsigned long previousMillisDisplay = 0;
int sudut = 0;
int slot = 2;
int isCarDetected = 0;
int mode = 2;
bool flag = true;
```

```
bool tag = true;
bool tagDelay = false;
bool first = true;
bool firstClose = true;
String status = "";
int temp = 0;
int used = 0;
int slotAll = 0;
int nonktp = 0;
String special = "false";

long debouncing_time = 15; //Debouncing Time in Milliseconds
volatile unsigned long last_micros;

Servo servo;
TM1637Display display(CLK, DIO);

void setup() {
  // Open serial communications and wait for port to open:

  Serial.begin(4800);
  mySerial.begin(4800);
  pinMode(D2, INPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(D4),    my_interrupt_handler,
      FALLING);
  pinMode(D7, INPUT);
  servo.attach(D6);

  servo.write(0);
  display.clear();
  display.setBrightness(0x0f);
  delay(2000);
```

```
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() { // run over and over

  special = Firebase.getString("special");
  slot = Firebase.getInt("slot");
  slotAll = Firebase.getInt("slotAll");
  nonktp = Firebase.getInt("nonktp");
  if (special=="true"){
    mode=3;
  }
//  Serial.println("IDLE");
  //Serial.println(digitalRead(D4));

  switch (mode){
    case 1 :
      //
      // Kirim UID ke database
      //
      showSlotPark(151,false);
      while (mySerial.available()>0) {
        mySerial.readBytes(value,7);
```

```
        if
      ((value[0]+value[1]+value[2]+value[3]+value[4]+value[5]+value[6
      ])>0){
          temp                                                    =
      value[0]+value[1]+value[2]+value[3]+value[4]+value[5]+value[6];
        } else first = true;

//        Serial.print("MODE ISI BACA KTP. Nilainya = ");
//        Serial.println(temp);

        if (temp>0){
          if (first) {
            first = false;

      Firebase.setInt("newUID",(value[0]+value[1]+value[2]+value[3]+v
      alue[4]+value[5]+value[6]));
          }

          //Firebase.setInt("UID",
      value[0]+value[1]+value[2]+value[3]+value[4]+value[5]+value[6])
      ;
        // handle error
          if (Firebase.failed()) {
              Serial.print("setting /message failed:");
              Serial.println(Firebase.error());
              return;
          }
        }
      }

      //Serial.println(Firebase.getString(String(value[0]+value[1]+va
      lue[2]+value[3]+value[4]+value[5]+value[6])+"/status"));
      first = true;
      break;
    case 2:
```

```
  //
  // Mobil terdeteksi di portal dan menunggu scan kartu
  //
  if (digitalRead(D7)==0 && slot>0){
    while (mySerial.available()>0) {
      mySerial.readBytes(value,7) ;
      //
      // Validasi ID sesuai database
      //
      if
((value[0]+value[1]+value[2]+value[3]+value[4]+value[5]+value[6
])>0){
        temp                                                    =
value[0]+value[1]+value[2]+value[3]+value[4]+value[5]+value[6];
      }
//          for (int i=0; i<7;i++){
//            Serial.print(value[i],HEX);
//            Serial.print(" ");
//          }
//          Serial.println();

//          Serial.print("MODE RUNNING BACA KTP. Nilainya = ");
//          Serial.println(temp);

      if (temp>0){
        status                                                  =
Firebase.getString("user/"+String(value[0]+value[1]+value[2]+va
lue[3]+value[4]+value[5]+value[6])+"/status");

        //Serial.println(status);
        if (!Firebase.failed()){
          if (status == "tidakparkir"){
            //Serial.println("Mau buka");
            openGate(); // Buka gerbang
```

```
            firstClose = true;

Firebase.setString("user/"+String(value[0]+value[1]+value[2]+va
lue[3]+value[4]+value[5]+value[6])+"/status", "parkir");
            if (tag){
              tag =false;
              tagDelay = true;
            }
          }
        } else {
          value[0] = 0x00; value[1] = 0x00; value[2] = 0x00;
          value[3] = 0x00; value[4] = 0x00; value[5] = 0x00;
          value[6] = 0x00;
        }
      }
    }
  } else{
    //
    // Untuk delay sebelum portal tertutup
    //
    //Serial.print("MODE RUNNING TIDAK ADA KARTU");
    if (tagDelay){
      previousMillisServo = millis();
      tagDelay = false;
      slot--;
      used = slotAll - slot;
      Firebase.setInt("slot",slot);
      Firebase.setInt("used",used);
    }
    if (firstClose) {
      firstClose = false;
      if (temp>0){
        String                          nama                          =
Firebase.getString("user/"+String(temp)+"/nama");
        Firebase.setString("parking/"+String(temp),nama);
```

```
        }
      }
      closeGate();
      tag = true;
    }
    showSlotPark(slot,false);
    //Serial.println(digitalRead(D4));

  break;
case 3 :
  //
  // Input Manual Plat Nomor
  //
  showSlotPark(0,true);

  if (digitalRead(D7)==0 && slot>0){
    openGate();
    tagDelay = true;
    firstClose=true;
  } else{
      //
      // Untuk delay sebelum portal tertutup
      //
      if (tagDelay){
        previousMillisServo = millis();
        tagDelay = false;
        slot--;
        used = slotAll - slot;
        nonktp++;
        Firebase.setInt("slot",slot);
        Firebase.setInt("used",used);
      }
      if (firstClose) {
        firstClose = false;
        String plat = Firebase.getString("inputmanual");
```

```
            Firebase.setString("parking/"+plat,"Anonymous");
            Firebase.setString("inputmanual","");
            Firebase.setInt("nonktp",nonktp);
            special = "false";
            Firebase.setString("special","false");
            mode=2;
          }
          closeGate();

          tag = true;
        }

      break;
  }


}

void showSlotPark(int slot, bool logic){
  if((millis() - previousMillisDisplay >= 1000))
  {
    previousMillisDisplay = millis();
  }
  display.showNumberDec(slot, logic);
}

void openGate(){
  if(millis() - previousMillisServo >= 2000)
  {
    previousMillisServo = millis();
    servo.write(90);
    //Serial.println("Udah buka");
  }
}
```

```
void closeGate(){
  if(millis() - previousMillisServo >= 2000)
  {
    previousMillisServo = millis();
    servo.write(0);
    //Serial.println("Udah tutup");
  }
}

void my_interrupt_handler()
{
 static unsigned long last_interrupt_time = 0;
 unsigned long interrupt_time = millis();
 // If interrupts come faster than 200ms, assume it's a bounce and
      ignore
 if (interrupt_time - last_interrupt_time > 1000)
 {
   mode++;
    if (mode>2) mode=1;
    Serial.println(mode);
 }
 last_interrupt_time = interrupt_time;
}

void debounceInterrupt() {
  if((long)(micros() - last_micros) >= debouncing_time * 1000) {
    handleInterrupt();
    last_micros = micros();
  }
}

void handleInterrupt() {
    mode++;
    if (mode>2) mode=1;
    //Firebase.setInt("mode",mode);
```

```
    // handle error
//    if (Firebase.failed()) {
//        Serial.print("setting /number failed:");
//        Serial.println(Firebase.error());
//        return;
//    }
  Serial.println(mode);
}
```