

LAMPIRAN A

Listing Program Arduino

```
#include <Nextion.h>
Nextion myNextion(Serial1, 9600);

String readid="";

#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 49
#define SS_PIN 53
#define BUZZ_PIN 12

MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance

String data="";
String status="";

unsigned long previousMillis = 0;
unsigned long currentMillis = 0;
unsigned long previousMillis1 = 0;
String barcode="";
String read_data="";
int timeout=6;
int timer_gagal=0;

void data_display(){
  data="ID:";
  data+=readid;
  data+=" CODE:";
  data+=barcode;
  data+=" STS:";
  data+=status;
  myNextion.setTextComponentText("data", data);
}
```

```

void read_barcode(){
  while(Serial2.available()){
    char c = Serial2.read();
    if(c!=13)read_data+=c;
    timeout=0;
  }

  if (currentMillis - previousMillis >= 100) {
    previousMillis = currentMillis;
    if(timeout<5)timeout++;
    if(timeout==4){
      barcode=read_data;
      data_display();
      Serial.print("BARCODE:");
      Serial.print(barcode);
      Serial.println();
      Serial3.print(" ");
      Serial3.print(barcode);
      Serial3.print("# ");
      delay(1000);
      read_data="";
    }
  }
}

```

```

void cek_fedback(){
  if(Serial3.available()){
    char c = Serial3.read();
    if(c=='O'){
      Serial.println("proses berhasil");
      myNextion.setText("tuliskan", "Proses Berhasil");
      delay(3000);
      myNextion.setText("tuliskan", "Selamat Datang di
Perpustakaan");
    }
  }
}

```

```

readid = "";
barcode = "";
status = "";
data_display();
timer_gagal=0;
}
if(c=='E'){
    Serial.println("id tidak terdaftar");
    myNextion.setText("tulis", "ID Tidak Terdaftar!");
    digitalWrite(BUZZ_PIN,1);
    delay(3000);
    digitalWrite(BUZZ_PIN,0);
    myNextion.setText("tulis", "Selamat Datang di
Perpustakaan");
    readid = "";
    barcode = "";
    status = "";
    data_display();
    timer_gagal=0;
}
}
}
void setup() {
    // konfigurasi I/O
    pinMode(BUZZ_PIN,OUTPUT);

    Serial.begin(9600); // Initialize serial communications
    myNextion.init();
    Serial2.begin(9600);
    Serial3.begin(9600);

    SPI.begin(); // Init SPI bus
    mfr522.PCD_Init(); // Init MFRC522 card
    Serial.println(" ");

    digitalWrite(BUZZ_PIN,1);

```

```

delay(200);
digitalWrite(BUZZ_PIN,0);
myNextion.setText("tulis", "Selamat Datang di
Perpustakaan");
}

```

```

void loop() {
String message = myNextion.listen();
if(message != ""){
Serial.println(message);
if (message == "65 0 5 0 ffff ffff ffff") { // pinjam
digitalWrite(BUZZ_PIN,1);
status = "meminjam";
Serial.print("STS:");
Serial.print(status);
Serial.println();
Serial3.print(" ");
Serial3.print(status);
Serial3.println("$ ");
data_display();
myNextion.setText("tulis", "Meminjam");
delay(100);
}
if (message == "65 0 3 1 ffff ffff ffff") { // perpanjang
digitalWrite(BUZZ_PIN,1);
status = "perpanjang";
Serial.print("STS:");
Serial.print(status);
Serial.println();
Serial3.print(" ");
Serial3.print(status);
Serial3.println("$ ");
data_display();
myNextion.setText("tulis", "Perpanjang");
delay(100);
}
}
}

```

```

}
if (message == "65 0 6 0 ffff ffff ffff") { // kembalikan
digitalWrite(BUZZ_PIN,1);
status = "mengembalikan";
Serial.print("STS:");
Serial.print(status);
Serial.println();
Serial3.print(" ");
Serial3.print(status);
Serial3.println("$ ");
data_display();
myNextion.setText("tulis", "Mengembalikan");
delay(100);
}
digitalWrite(BUZZ_PIN,0);
}

```

```

currentMillis = millis();

```

```

if (currentMillis - previousMillis >= 1000) {
  previousMillis = currentMillis;
  if(status!=""&&readid!=""&&barcode!=""){
    timer_gagal++;
    if(timer_gagal>10){
      Serial.println("proses gagal");
      myNextion.setText("tulis", "Proses Gagal!");
      digitalWrite(BUZZ_PIN,1);
      delay(3000);
      digitalWrite(BUZZ_PIN,0);
      myNextion.setText("tulis", "Selamat Datang di
Perpustakaan");
      readid = "";
      barcode = "";
      status = "";
      data_display();
      timer_gagal=0;
    }
  }
}

```

```

    }
    }

}

read_barcode();

cek_feedback();

digitalWrite(BUZZ_PIN,0);
if ( ! mfr522.PICC_IsNewCardPresent() || !
mfr522.PICC_ReadCardSerial() ) {

    return;
}

digitalWrite(BUZZ_PIN,1);
readid="";
for (byte i = 0; i < mfr522.uid.size; i++) {
    readid += String(mfr522.uid.uidByte[i]);
}
Serial.print("ID:");
Serial.print(readid);
Serial.println();
Serial3.print(" ");
Serial3.print(readid);
Serial3.println("@ ");
data_display();
delay(200);
digitalWrite(BUZZ_PIN,0);

delay(1000);

}

```

LAMPIRAN B

Listing Program ESP32

```
// json 6.18.5
#include <ArduinoJson.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <WiFiMulti.h>
#include <WiFiClient.h>

WiFiClient client;

const char* ssid = "mywifi";
const char* password = "354354354";

String ip_server = "http://192.168.1.15/";
String serverName = ip_server + "rfidktp/write-data.php";
StaticJsonDocument<200> doc;

unsigned long previousMillis = 0;
const long interval = 100;

#define pinled 2

String status_="";
String barcode = "";
String readid="";
String indata="";

void baca_data(){
  if(Serial2.available()){
    char c = Serial2.read();
    if(c=='@'||c=='#'||c=='$'){ }
    else{
      indata+=c;
    }
    if(c==' ')indata="";
  }
}
```

```

if(c=='@'){
readid="";
readid = indata;
Serial.print("ID:");
Serial.println(readid);
indata="";
}
if(c=='#'){
barcode = "";
barcode = indata;
Serial.print("BARCODE:");
Serial.println(barcode);
indata="";
}
if(c=='$'){
status_="";
status_ = indata;
Serial.print("STATUS:");
Serial.println(status_);
indata="";
}
}
}

void setup() {

pinMode(pinled, OUTPUT);

// Initialize serial port
Serial.begin(9600);
Serial2.begin(9600);

Serial.print("Connecting to ");
Serial.print("SSID:");
Serial.print(ssid);
Serial.print(" PASS:");

```



```

Serial.println(password);
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
    digitalWrite( pinled, digitalRead( pinled ) ^ 1 );
}

Serial.println("");
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
Serial.print("IP Server: ");
Serial.println(ip_server);
Serial.println("Ready");

}

void loop() {

    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        digitalWrite( pinled, digitalRead( pinled ) ^ 1 );
    }

    baca_data();

    if(status_!=""&&readid!=""&&barcode!=""){
        baca_database(readid);
        status_="";
        readid="";
        barcode="";
    }
}

```

```

    //test_manual();
}

String in="";
void test_manual(){

    if(Serial.available()){
        char c = Serial.read();
        if(c!='#'){
            in+=c;
        }
        if(c=='#'){
            Serial.println(in);
            tulis_database(in,"pinjam","1234");
            in="";
        }
    }
}

void baca_database(String pемbanding){

    if ((WiFi.status() == WL_CONNECTED)) { //Check the current
connection status

        String endpoint = ip_server + "rfidktp/api.php?uid=";
        endpoint += pемbanding;

        HTTPClient http;
        http.begin(client,endpoint); //Specify the URL
        http.GET(); //Make the request
        String payload = http.getString();
        Serial.println(payload.length());
        Serial.println(payload);
    }
}

```

```

DeserializationError error = deserializeJson(doc, payload);

// Test if parsing succeeds.
if (error) {
  digitalWrite( pinled, 0 );

  Serial.print(F("deserializeJson() failed: "));
  Serial.println(error.f_str());
  return;
}

String data_terima = doc[0]["uid"];
Serial.println(data_terima);
if(data_terima == pемbanding){
Serial.println("UID TERDAFTAR");
Serial.println(status_);
tulis_database(readid, status_,barcode);
Serial2.print("O");
}
else{
Serial.println("UID TIDAK TERDAFTAR");
Serial2.print("E");
}
}

void baca_database_realtime(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    baca_database("xxxx");
  }
}

void tulis_database(String _rfid, String _status, String _barcode){

```

```

//Check WiFi connection status
if(WiFi.status()== WL_CONNECTED){
  WiFiClient client;
  HTTPClient http;
  Serial.println("send database");
  String serverPath = serverName;
  serverPath += "?rfid=";
  serverPath += _rfid;
  serverPath += "&status=";
  serverPath += _status;
  serverPath += "&barcode=";
  serverPath += _barcode;

  // Your Domain name with URL path or IP address with path
  http.begin(client, serverPath.c_str());

  // Send HTTP GET request
  int httpStatusCode = http.GET();

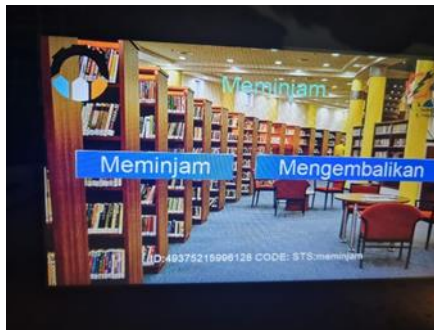
  if (httpStatusCode>0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpStatusCode);
    String payload = http.getString();
    Serial.println(payload);
  }
  else {
    Serial.print("Error code: ");
    Serial.println(httpStatusCode);
  }
  // Free resources
  http.end();
}
else {
  Serial.println("WiFi Disconnected");
}
}

```

LAMPIRAN C
Hasil Alat dan Pengujian



Gambar (a) Proses peminjaman



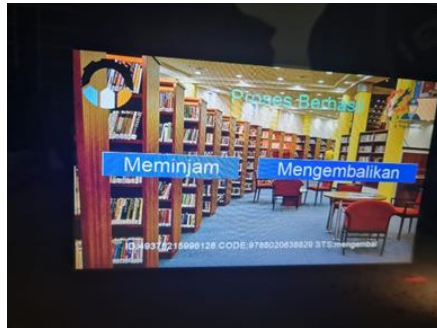
Gambar (b) Status Meminjam



Gambar (c) Proses Peminjaman Berhasil



Gambar (d) Proses Mengembalikan



Gambar (e) Proses Mengembalikan Berhasil



Gambar (f) Hasil Tap KTM