Coding Menggunakan *software* Arduino:

```
#include <Servo.h>

#define DEBUG true

#define esp8266 Serial3

#define pintu 25

#define led1 10

#define led2 11

#define led3 12

#define led4 13

#define buka 165

#define tutup 45

#define reset_time 30000

long waktu_proses = 0;

Servo sPintu;

String pesan_masuk;

//SoftwareSerial esp8266(2,3); // make RX Arduino line is pin 2, make TX Arduino line is pin 3.

// This means that you need to connect the TX line from the esp to the Arduino's pin 2

// and the RX line from the esp to the Arduino's pin 3

void setup()

{

  Serial.begin(9600);

  esp8266.begin(115200); // your esp's baud rate might be different

  pinMode(led1, OUTPUT);
  digitalWrite(led1, 1);

  pinMode(led2, OUTPUT);

  digitalWrite(led2, 1);

  pinMode(led3, OUTPUT);

  digitalWrite(led3, 1);

  pinMode(led4, OUTPUT);

  digitalWrite(led4, 1 );

  sPintu.attach(pintu);

  sPintu.write(tutup);

  set_esp();

  pesan_masuk = "";

  waktu_proses = millis();

}

void set_esp() {

  sendData("AT+RST\r\n", 2000, DEBUG); // reset module

  sendData("AT+CWMODE=2\r\n", 1000, DEBUG); // configure as access point

  sendData("AT+CWSAP=\"MY HOUSE\",\"12345678\",3,3\r\n", 1000, DEBUG); //
configure as access point

  //sendData("AT+CIPAP=\"192.168.0.1\"r\n", 1000, DEBUG); // configure as
access point

  //sendData("AT+CIFSR\r\n", 1000, DEBUG); // get ip address

  sendData("AT+CIPMUX=1\r\n", 1000, DEBUG); // configure for multiple
connections

  sendData("AT+CIPSERVER=1,80\r\n", 1000, DEBUG); // turn on server on port
80

}
```

```
void loop()

{

  if (millis() > waktu_proses + reset_time) {

    //set_esp();

    waktu_proses = millis()

  }

  if (esp8266.available()) // check if the esp is sending a message

  {

    if (esp8266.find("+IPD,"))

    {

      delay(1000); // wait for the serial buffer to fill up (read all the
serial data)

      // get the connection id so that we can then disconnect

      int connectionId = esp8266.read() - 48; // subtract 48 because the
read() function returns

      // the ASCII decimal value and 0 (the first decimal number) starts at
48


      esp8266.find("?"); // advance cursor to "pin="

      delay(1000);

      pesan_masuk = "";

      while (1) {

        char a =  (char)esp8266.read();

        if (a != ' ') {
```

```
    pesan_masuk += a;
  } else {
    Serial.println(pesan_masuk);
    if (pesan_masuk.indexOf("led1=1") != -1) {
      digitalWrite(led1, 0);
    }
    if (pesan_masuk.indexOf("led1=0") != -1) {
      digitalWrite(led1, 1);
    }

    if (pesan_masuk.indexOf("led2=1") != -1) {
      digitalWrite(led2, 0);
    }
    if (pesan_masuk.indexOf("led2=0") != -1) {
      digitalWrite(led2, 1);
    }

    if (pesan_masuk.indexOf("led3=1") != -1) {
      digitalWrite(led3, 0);
    }
    if (pesan_masuk.indexOf("led3=0") != -1) {
      digitalWrite(led3, 1);
    }
    if (pesan_masuk.indexOf("led4=1") != -1) {
      digitalWrite(led4, 0);
    }
    if (pesan_masuk.indexOf("led4=0") != -1) {
      digitalWrite(led4, 1);
    }

    if (pesan_masuk.indexOf("pintu=1") != -1) {
      sPintu.write(buka);
    }

    if (pesan_masuk.indexOf("pintu=0") != -1) {
      sPintu.write(tutup);
    }

    break;
  }
}
// make close command
String isi = "{\"success\":1}";
String balasan = "AT+CIPSEND=";
balasan += connectionId; // append connection id
balasan += ",13\r\n";
sendData(balasan, 1000, DEBUG); // close connection
```

```
      //delay(2000);
      sendData(isi+"\r\n", 1000, DEBUG); // close connection
      //delay(3000);
      String closeCommand = "AT+CIPCLOSE=";
      closeCommand += connectionId; // append connection id
      closeCommand += "\r\n";
      sendData(closeCommand, 1000, DEBUG); // close connection
    }
  }
}
/*
* Name: sendData
* Description: Function used to send data to ESP8266.
* Params: command - the data/command to send; timeout - the time to wait for
a response; debug - print to Serial window?(true = yes, false = no)
* Returns: The response from the esp8266 (if there is a reponse)
*/
String sendData(String command, const int timeout, boolean debug)
{
  String response = "";
  esp8266.print(command); // send the read character to the esp8266
  long int time = millis();
  while ( (time + timeout) > millis())
  {
    while (esp8266.available())
    {
      // The esp has data so display its output to the serial window
      char c = esp8266.read(); // read the next character.
      response += c;
    }
  }
  if (debug)
  {
    Serial.print(response);
  }
  return response;
}
```