

## DAFTAR PUSTAKA

- [1] Lukar, T. Y. H. ., & Setiawan, F. B. *Deteksi Sinyal Otot Manusia Pada Android Menggunakan Sensor Elektromiografi Berbasis Mikrokontroler Arduino Uno*. 99–106. (2019). <https://doi.org/10.5614/sniko.2018.15>
- [2] Ivan, V., & Wahab, F. *Pendeteksian Sinyal Otot Lengan Manusia Menggunakan Sensor Otot EMG Berbasis Arduino Uno*. 76–80. (2020).
- [3] Sulistyawati, I. N., & Kholis, N. RANCANG BANGUN ELEKTROMIOGRAF (EMG) BERBASIS MIKROKONTROLER UNTUK MENDETEKSI CEDERA OTOT PADA PERGELANGAN KAKI (ANKLE) Ima Noviana Sulistyawati. *Jurnal Teknik Elektro, VIII*, 557–562, (2018).
- [4] Raharjo, A. B., Fatukhurrozi, B., & Nawawi, I. Analisis sinyal electromyography (emg) pada otot biceps brachii untuk mendeteksi kelelahan otot dengan metode median frekuensi. *Journal of Electrical Engineering, Computer and Information Technology*, 1(1), 1–5, (2020). <https://jurnal.untidar.ac.id/index.php/thetaomega/article/view/3046/1409%0A>  
<https://jurnal.untidar.ac.id/index.php/thetaomega/article/view/3046>
- [5] F., Rofii, F., & Setiawidayat, S. Rancang Bangun Alat Terapi Lengan Continous Passive Motion (CPM) dengan Control Electromyograph (EMG) Untuk Pasien Pasca Operasi dan Stroke. *Teknik*, 40(3), 176, (2019). <https://doi.org/10.14710/teknik.v40i3.25119>
- [6] Ubaidillah, M. J., Munadhif, I., & Rinanto, N. Klasifikasi Gelombang Otot Lengan Pada Robot Manipulator Menggunakan Support Vector Machine. *Rekayasa*, 12(2), 91–97, (2019) <https://doi.org/10.21107/rekayasa.v12i2.5406>
- [7] Gunawan, H. Y., & Setiawan, F. B. *Perancangan Penampil Grafik Sinyal Ketegangan Otot Perut Dengan Menggunakan Sensor Elektromiografi*, 10–11, (2019). <https://doi.org/10.5614/sniko.2018.14>

## LAMPIRAN

Kode program berfungsi untuk menjalankan sensor sesuai dengan fungsi yang diberikan. NodeMCU bertindak sebagai pengirim data ke *platform* ubidots. Kode program dapat dilihat pada gambar 3.5 sampai 3.9.

```
#define SENSOR A0
```

**Gambar 3.5 Mengatur Sensor menjadi A0**

```
float nilai = analogRead(SENSOR);  
float eng = nilai*(5.0/1024);
```

**Gambar 3.6 Kode Program Sensor V3**

```
#include <ESP8266WiFi.h>  
#include <PubSubClient.h>  
#include <UbidotsESP8266.h>
```

**Gambar 3.7 Library pada NodeMCU**

```

void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFISSID, PASSWORD);
  // Assign the pin as INPUT
  pinMode(SENSOR, INPUT);

  Serial.println();
  Serial.print("Waiting for WiFi...");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  client.setServer(mqttBroker, 1883);
  client.setCallback(callback);
}

```

**Gambar 3.8 Kode Setup NodeMCU**

```

void loop() {
  if (!client.connected()) {
    reconnect();
  }

  sprintf(topic, "%s", "/v1.0/devices/", DEVICE_LABEL);
  sprintf(payload, "%s", ""); // Clears the payload
  sprintf(payload, "%s", VARIABLE_LABEL); // Adds the variable label
  float nilai = analogRead(SENSOR);
  float avg = nilai*(5.0/1024);

  // 4 is decimal width, 2 is precision! Float value to copied onto str_sensor
  dtostrf(avg, 4, 2, str_sensor);

  sprintf(payload, "%s (%s): %s", topic, str_sensor); // Adds the value
  Serial.println("Publishing data to Ubidots Cloud");
  Serial.println(avg);
  client.publish(topic, payload);
  client.loop();
  delay(100);
}

```

**Gambar 3.9 Kode *loop* Setup NodeMCU**