DAFTAR PUSTAKA

Achlison, U., & Bambang, S. 2020. Analisis Hasil Ukur Sensor Load Cell untuk Penimbang Berat Beras, Paket dan Buah berbasis Arduino. Jurnal Ilmiah Ekonomi Dan Bisnis. 13(1).

Adiprasetyo, N. F. Alat Pemberi Makan Kucing Otomatis Berbasis Arduino Uno. Fakultas Teknik. Universitas Muhammadiyah Malang. 2017.

Arnida, Andi. *Perbedaan Efektivitas Kompres Hangat dan Kompres Normal Salin pada Skala Phlebits di RSUD Labuang Baji Makassar*. 2015. Undergraduate (S1) thesis, Universitas Islam Negeri Alauddin Makassar.

Ferdiansyah M.A, dkk. 2018. *Perancangan Sistem Kontrol Intensitas Lampu Ruang Kuliah dan Kontrol Suhu untuk Efisiensi Daya Menggunakan Arduino Uno*. Jember: Universitas Muhammadiyah Jember.

Fina S., dkk Perancangan Robot Pencapit Untuk Penyortir Barang Berdasarkan Warna LED RGB Dengan Display LCD Berbasis Arduino Uno. Universitas Mercubuana. Jakarta Barat. 2014.

Giant, Ragil Febrio; Darjat dan Sudjadi, "Perancangan Aplikasi Pemantau Dan Pengendali Piranti Elektronik Pada Ruangan Berbasis Web," p. 3, 2015.

Handoyono, Triyanto, E. dan Latifah L. Hubungan Pengetahuan tentang Perawatan Terapi Intravena dengan Angka Kejadian Plebitis Di RSUD Prof Dr. Margono Soekardjo Purwokerto. *Soedirman Nursing Journal* .2006.

Indoware. 2014. *User Manual timbangan loadcell HX711 Indoware Elektronik*. Semarang: Indo-ware Elektornik.

Iqbal, K. Desain Kontrol dan Monitoring *Smart Home* Berbasis Android. 2021

Komite Keperawatan Rumah Sakit Baptis Kediri, (2013). Standar Prosedur Operasional Menyiapkan dan Memberikan Infus.

Mardiati R, dkk. 2016. *Rancang Bangun Prototype Sistem Peringatan Jarak Aman pada Kendaraan Roda Empat Berbasis Mikrokontroler ATMEGA32*. Bandung: Universitas Islam Negeri Sunan Gunung Djati. 2(1).

[illegible], R. Sistem Monitoring dan Peringatan Pada Volume Cairan Intravena [illegible]s) Pasien Menggunakan Arduino Berbasis Website. Jurnal Komputer dan [illegible]kasi. Volume 07, No. 03. 2019.

Muchlis, N. R. Rancang Bangun Sistem Monitoring dan Kontroling Infus Menggunakan Mikrokontroler Berbasis Website. 2024.

Muhammad, Arie Kurniawan. 2016. Aplikasi Accelerometer pada Penstabil Monopod Menggunakan Motor Servo. http://eprints.polsri.ac.id. Diakses Tanggal 16 juni 2024.

Nuryanto, R.U. A. Sherwin, dan R.F. Robot, 2015. Rancang Bangun Otomatis Sistem Infus Pasien Universitas Sam Ratulangi. Manado.

Potter, P.A, Perry, A.G. Buku Ajar Fundamental Keperawatan : Konsep, Proses, dan Praktik. Edisi 4. Volume 2. Jakarta. 2005.

Pranjoto, H., dkk. Penetuan Cairan Infus Masuk ke Pasien Secara Otomatis Lewat Parameter Berat Menggunakan Jaringan Nirkabel. Universitas Katolik Widya Mandala. Surabaya. 2019.

Purnamasari, Putri Indraningtyas, (2013). Hubungan Lama Pemasangan Infus Dengan Kejadian Plebitis di RSUD Tugurejo Semarang. Jurnal Keperawatan STIKES Telogorejo Semarang, Jawa Tengah.

Rosyidi Sa'ad, dkk. 2019. *Rancang Bangun Alat Pembersih Dan Penyortir Ukuran Telur Asin Berbasis Arduino Mega 2560*. Malang: Institut Teknologi Nasional.

Sides, C. R., Li, N., Patrissi, C. J., Scrosati, B., & Martin, C. R. (2002). Nanoscale materials for lithium-ion batteries. In MRS Bulletin (Vol. 27, Issue 8, pp. 1–4). https://doi.org/10.1557/mrs2002.195

Sucipta Indra, dkk. 2021. *Prototype Pemantauan Tetesan Cairan Infus Berbasis IoT Terkoneksi Perangkat Android*. Jurnal Teknik Elektro Universitas Telkom 12(3).

Sugiyono, 2009, Metode Penelitian Kuantitatif, Kualitatif dan R&D, Bandung : Alfabeta.

T. S. Prayogo, "TA: Sistem Kontrol Charger Handphone Otomatis Berbasis Android," 2016.

LAMPIRAN

Lampiran 1

```
/*
   attachInterrupt(0, rpm_fun, FALLING);
  LOW = interrupt akan dieksekusi terus menerus selama pin membaca logika
LOW
  RISING = LOW ke HIGH
  CHANGE = HIGH ke LOW ataupun LOW ke HIGH
  FALIING = HIGH ke LOW
*/
#include <SoftwareSerial.h>
#include <Arduino.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
// #include <TimerOne.h>
#include <Wire.h>
#include <HX711.h>
#define infusKnPin 2
#define buzzerPin 13
#define DOUT 5
#define CLK 4
const int button1Pin = 8; // Ganti pin sesuai dengan koneksi tombol
const int button2Pin = 9;
const int button3Pin = 10;

unsigned long jumlahKn = 0;
unsigned long oldJumlahKn = 0;
bool stringComplete = false;
float calibration_factor = 470;  //Hasil Kalibrasi  470.00
float units;
int pos = 30;
String data;
char c;
HX711 scale;
byte beepSt = 0;
Servo myservo;              //
SoftwareSerial uno(11, 12);  //RX,TX
LiquidCrystal_I2C lcd(0x27, 16, 2);

void counterKn() {
  Kn = millis();


p() {
```

```
  Serial.begin(9600);
  uno.begin(9600);
  pinMode(infusKnPin, INPUT_PULLUP);
  pinMode(buzzerPin, OUTPUT);

  // Kecepatan
  attachInterrupt(digitalPinToInterrupt(infusKnPin), counterKn, FALLING);  //
RISING);
  myservo.attach(6);
  myservo.write (30); // keadaan awal posisi 30 derajat
  lcd.init();
  lcd.backlight();  //Depan Belakang //Servo Kanan
  scale.begin(DOUT, CLK);
  scale.set_scale();
  scale.tare();                         //Reset the scale to 0
  long zero_factor = scale.read_average();  //Get a baseline reading
  Serial.print("Zero factor: ");          //This can be used to remove the need to tare
the scale. Useful in permanent scale projects.
  Serial.println(zero_factor);
  delay(3000);
}
void loop() {
  terima();
  // Target();
  hitungKn();

  tampilan();
  terima();
  // beep();
}
```

Lampiran 2

```
int countKn;
int target;
unsigned long previousMicros = 0;
const unsigned long interval = 5000000;
int toleransi = 5;  //toleransi dari target infus
float selisihKn;
float TetesPerMenitKn;
      long oldKn;
      arget;
```

```
void hitungKn() {
  unsigned long tundaKn = millis();
  if (tundaKn - oldKn > 250) {
    oldKn = tundaKn;
    if (oldJumlahKn != jumlahKn) {
      selisihKn = jumlahKn - oldJumlahKn;
      oldJumlahKn = jumlahKn;
      //beepSt = 1;
    }
    countKn++;
    if (countKn % 4 == 0) {
      // Serial.print("jumlah = ");
      // Serial.print(jumlahKn);
      // Serial.print(", waktPerTetes = ");

      //WAKTU PER TETESAN
      float waktPerTetesKn = selisihKn / 1000;

      // Serial.print(waktPerTetesKn, 1);  //,DEC);
      // Serial.print(", TetesPerMenit = ");
      float oldTetesPM;

      // RUMUS KECEPATAN TETESAN
      TetesPerMenitKn = 60 / waktPerTetesKn;

      if (TetesPerMenitKn < 300) {
        oldTetesPM = TetesPerMenitKn;
      } else TetesPerMenitKn = oldTetesPM;

      // JIKA TETESAN LEBIH DARI TARGET
      if (TetesPerMenitKn > (target + toleransi)) {
        pos--;  //POSISI SERVO BERKURANG 1 DERAJAT
      }

      // JIKA TETESAN KURANG DARI TARGET
      if (TetesPerMenitKn < (target - toleransi)) {
        pos++;  //POSISI SERVO BERTAMBAH 1 DERAJAT
      }

      pos = constrain(pos, 0, 155);
      myservo.write(pos);
      terima();
                );
                ();
```

```
void berat() {                    //MENGHITUNG BERAT
  scale.set_scale(calibration_factor);  //MENGKALIBRASI NILAI BERAT
AWAL
  units = scale.get_units(), 1;
  if (units < 0) {
    units = 0.00;
  }
  units;
}
void kirim() {  //POST DATA
  unsigned long currentMicros = micros();
  // MENGHITUNG SELAMA 5 DETIK KEMUDIAN MENGIRIM DATA KE
ESP
  if (currentMicros - previousMicros >= interval) {
    String TPM = String(TetesPerMenitKn, 1);
    String berat = String(units, 1);
    uno.print(TPM);  //MENGIRIM DATA KECEPATAN TETESAN
    uno.print(" ");
    uno.println(berat);  //MENGIRIM DATA BERAT
    Serial.print("TPM = ");
    Serial.print(TetesPerMenitKn, 1);  //,DEC);
    Serial.print(", Berat: ");
    Serial.print(units);
    Serial.print(" Gram");
    Serial.print(", servo = ");
    Serial.print(pos);
    Serial.println();
    previousMicros = currentMicros;
  }
}
void tampilan() {       // MENAMPILKAN NILAI PADA LCD
  lcd.setCursor(0, 0);  // MENGATUR TAMPILAN UNTUK BERIS PERTAMA
  lcd.print("Kec: ");
  lcd.print(TetesPerMenitKn, 1);
  lcd.print(" ->");
  lcd.print(target);
  lcd.print("-");
  lcd.setCursor(0, 1);  //MENGATUR TAMPILAN UNTUK BARIS KEDIA
  lcd.print("Berat: ");
  lcd.print(units, 1);
}

na() {  // FUNGSI PENERIMA DATA DARI ESP
no.available()) {
10);
.read();  //MENERIMA DATA TARGET DARI ESP
t += d;
```

```
    target = Ntarget.toInt();
    Serial.print("target:");
    Serial.println(target);
  }
  Ntarget = "";
}
```

Lampiran 3

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>

int buzzerPin = 14;
SoftwareSerial node(D1, D2);  // RX, TX
// Variabel baca data
String data = "";
char c;
const char* ssid = "Arduino";
const char* password = "12345679";
int Target = 0;
int toleransi = 5;
int alarm = 0;

// pengenal host (server) = IP Address komputer server
const char* host = "http://smartinfusunhas.com/infusions";
int cepat, berat;
int alert = 0;
String idMac = "7070";
String target;

const char* WebURL = "http://smartinfusunhas.com/api/infusions";
const char* mac_system;

int id_new = 0;
short perintah = 3;

void setup() {
  Serial.begin(9600);
  node.begin(9600);
  e(buzzerPin, OUTPUT);
  gin(ssid, password);
  Serial) {
  for serial port to connect. Needed for native USB port only
```

```
  }

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Menghubungkan ke WiFi...");
  }
  Serial.println("Terhubung ke WiFi");
}

void loop() {
  digitalWrite(buzzerPin, LOW);
  while (node.available()) {  //MENERIMA DATA DARI ARDUINO
    delay(10);
    c = node.read();
    data += c;
  }
  if (data.length() > 0) {
    Serial.print("Data: ");
    web();        // GET ATAU AMBIL DATA DARI WEBSITE
    pisahData();  // PISAH DATA YANG DITERIMA DARI ARDUINO DARI 1
KALIMAT MENJADI 2 DATA (KECEPATAN DAN BERAT)
    kirim();      // KIRIM DATA KE WEBSITE
  }
  data = "";
}
//MEMISAH DATA YANG DITERIMA DARI ARDUINO
void pisahData() {
  // Mencari posisi spasi dalam string
  int spacePos = data.indexOf(' ');
  if (spacePos != -1) {
    // Jika spasi ditemukan, mengurai string
    cepat = data.substring(0, spacePos).toInt();   //MEMISAH DATA
KECEPATAN TPM
    berat = data.substring(spacePos + 1).toInt();  //MEMISAH DATA BERAT
INFUS
    Serial.print("Kecepatan: ");
    Serial.print(cepat);
    Serial.print(", Berat: ");
    Serial.println(berat);
    if (berat < 100) {  //JIKA BERAT KURANG DARI 100 MAKA BUZZER
DAN LED AKAN BERKEDIP
      digitalWrite(buzzerPin, HIGH);
      (100);
      lWrite(buzzerPin, LOW);
```

```
   //JIKA NILAI KECEPATAN TIDAK SAMA DENGAN TARGET TPM +
TOLERANSI, BANYAK NILAI SALAH YANG MASUK TERSEBUT DI
HITUNG
   if (cepat != (Target - toleransi) && cepat != (Target + toleransi)) {
     alarm++;           // JUMLAH NILAI YANG SALAH BERTAMBAH 1
     if (alarm >= 36) {  // JIKA NILAI SALAH BERTAMBAH TERUS
MENERUS HINGGAH 36 (BERTAMBAH SELAMA 3 MENIT) MAKA
BUZZER DAN LED AKAN BERKEDIP
       digitalWrite(buzzerPin, HIGH);
       delay(100);
       digitalWrite(buzzerPin, LOW);
       alert = 1;  // NILAI ALARM (VALUE ALERT) YANG DIKIRIM KE
WEBSITE
     }
     if (cepat >= (Target - toleransi) && cepat <= (Target + toleransi)) {  //JIKA
NILAI KECEPATAN SAMA DENGAN NILAI TARGET + TOLERANSI
MAKA:
       alarm = 0;                                          //JUMLAH NILAI YANG
SALAH TER RESET KE 0
       alert = 0;                                         //NILAI ALARM (VALUE
ALERT) YANG DIKIRIM KE WEBSITE
     }
   }

  } else {
    Serial.println("Tidak dapat menemukan spasi sebagai pemisah.");
  }
}
// POST ATAU MENGIRIM DATA KE WEBSITE
void kirim() {
  WiFiClient client;
  HTTPClient http;

  // Memulai HTTP client
  http.begin(client, WebURL);

  // Tentukan tipe konten (application/json)
  http.addHeader("Content-Type", "application/json");

  // Buat objek JSON
  StaticJsonDocument<200> jsonDoc;
  jsonDoc["laju_cairan"] = cepat;
          ["volume_infus"] = berat;
          ["mac"] = idMac;
          baru untuk alarm
          ["alert"] = alert;
```

```
  // Konversi JSON ke string
  String requestBody;
  serializeJson(jsonDoc, requestBody);

  // KIRIM DATA KE WEBSITE
  int httpResponseCode = http.POST(requestBody);

  // Cek kode respons
  if (httpResponseCode > 0) {
    String response = http.getString();  // Dapatkan respons dari server
    Serial.println("HTTP Response code: " + String(httpResponseCode));
    Serial.println("Response: " + response);
  } else {
    Serial.println("Error on sending POST: " + String(httpResponseCode));
  }

  // Akhiri koneksi HTTP
  http.end();
}

//GET
void web() {
  WiFiClient client;
  HTTPClient http;
  http.begin(client, host);   // menghubungkan ke server
  int httpCode = http.GET();  // mengirim permintaan pengambilan data
  if (httpCode == 200) {      // respon code 200 = disetujui
    if (httpCode == HTTP_CODE_OK) {
      String payload = http.getString();  // mengambil data dari database
      Serial.println("Data JSON diterima:");
      Serial.println(payload);
      DynamicJsonDocument doc(1024);  // Ubah sesuai kebutuhan
      deserializeJson(doc, payload);
      JsonArray data = doc.as<JsonArray>();  // Mengurai data JSON
      for (JsonObject item : data) {
        String macIn = item["mac"];         // MENGURAI DATA MAC
        int id = item["id"];                // MENGURAI DATA ID
        String target = item["target_tpm"];  // MENGURAI DATA TARGET TPM


        // mencocokkan data sesuai dengan perangkat
        if (id > id_new) {       // JIKA TERDAPAT DATA BARU
          macIn == idMac) {  // JIKA MAC PADA DATA BARU SAMA
          N MAC PERANGKAT MAKA:
          ode.print(target);  // MENGIRIM DATA TARGET KE ARDUINO
          erial.print("nilai target: ");
          erial.println(target);
```

```
          Target = target.toInt();
        }
      }
      id_new = id;  // UPDATE ID TERBARU
    }
  }
 } else {
  Serial.println("Gagal mengambil data dari endpoint.");
 }
 Serial.println(id_new);

 http.end();
 client.stop();
}
```