

# LAMPIRAN KARYA CIPTA

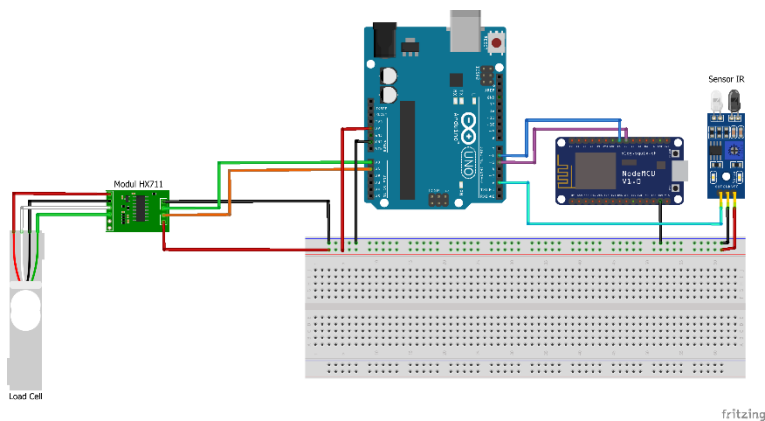
## BUKU PANDUAN

### A. Pengertian Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah sebuah aplikasi yang berfungsi untuk memprogram *board* Arduino Uno. Aplikasi ini bersifat *open source* dan mudah untuk digunakan (Galadima, 2014). Arduino IDE sendiri dapat dijalankan diberbagai sistem operasi seperti Windows, Mac OS, Linux. Aplikasi ini terus mengalami perbaikan dan terakhir diperbarui adalah versi Arduino 1.8.15. Arduino IDE dibuat dari bahasa pemrograman Java akan tetapi di dalam menuliskan program pada *sketch* yaitu program yang ditulis dengan menggunakan *software* Arduino digunakan bahasa yang mirip dengan bahasa pemrograman C.

### B. Sistem Pengkabelan

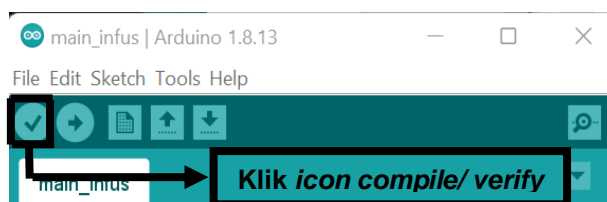
*Wiring* alat:



Gambar 1. Wiring sistem

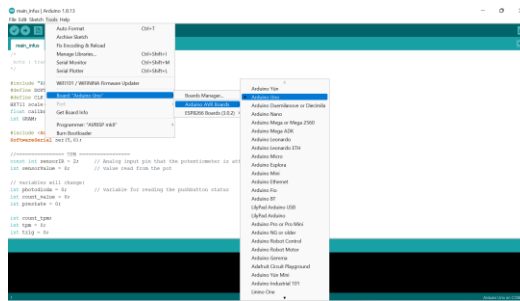
### C. Setup Arduino IDE dengan Arduino Uno (*Transmitter*)

1. Langkah pertama adalah buka aplikasi Arduino IDE. Lakukan penyusunan program di sketch kemudian *compile/ verify* program terlebih dahulu sebelum *upload* program pada Arduino Uno untuk melihat apakah pada program terdapat error.



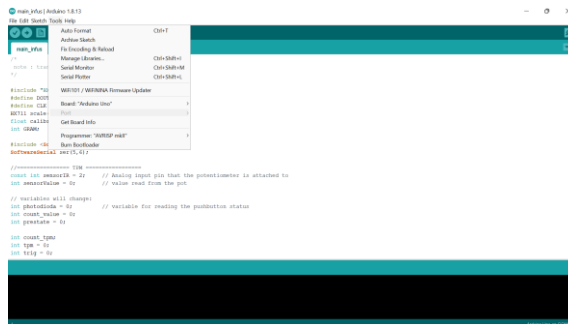
Gambar 2. Icon Compile/ Verivy

2. Pilih *board* yang sesuai dengan *board* yang digunakan, sesuaikan dengan pilihan berikut *Tools* → *Board* → *Arduino AVR Boards* → *Arduino Uno*



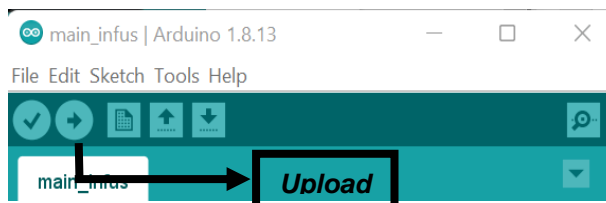
Gambar 3. Pemilihan Board

3. Setelah pemilihan *board* selesai dilakukan pada tahapan ini memilih port yang digunakan oleh *board* Arduino Uno. Untuk memastikan port yang digunakan oleh Arduino Uno lakukan dengan menekan tombol windows pada *keyboard* lalu mencari *device manager*, kemudian liat port yang terhubung akan ada keterangan Arduino atau bukan.
4. Setelah memastikan port yang digunakan oleh Arduino Uno maka selanjutnya dilakukan pemilihan port pada Arduino IDE, dengan dengan pilihan berikut *Tools* → *Port* → ...



Gambar 4. Pemilihan Port

5. Setelah memilih port yang sesuai dengan yang digunakan, Langkah terakhir adalah *upload* program.



Gambar 5. Upload Program

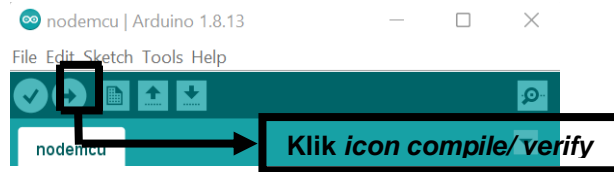
6. Jika sudah selesai dan proses uploading berhasil maka akan muncul tampilan seperti berikut,



Gambar 6. Berhasil Upload

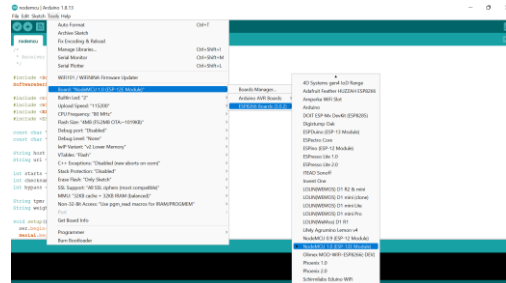
#### D. Setup Arduino IDE dengan NodeMCU ESP8266 (*Receiver*)

1. Langkah pertama adalah buka aplikasi Arduino IDE. Lakukan penyusunan program di sketch kemudian *compile/verify* program terlebih dahulu sebelum *upload* program pada Arduino Uno untuk melihat apakah pada program terdapat error.



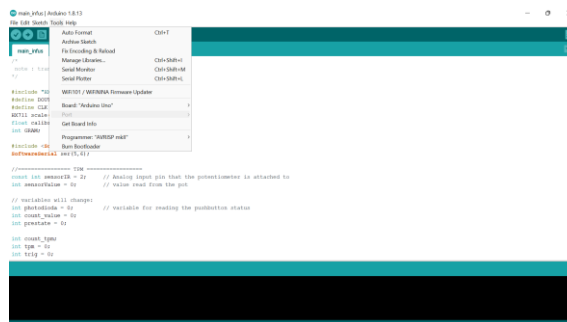
Gambar 7. Icon Compile/ Verivy

- Pilih *board* yang sesuai dengan *board* yang digunakan, sesuaikan dengan pilihan berikut *Tools* → *Board* → *ESP 8266 Boards* → *NodeMCU 1.0 (ESP-12E Module)*



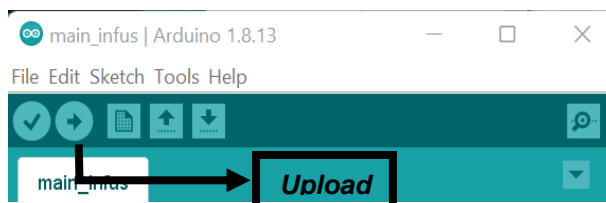
Gambar 8. Pemilihan Board

- Setelah pemilihan *board* selesai dilakukan pada tahapan ini memilih port yang digunakan oleh *board* NodeMCU ESP8266. Untuk memastikan port yang digunakan oleh NodeMCU ESP8266 lakukan dengan menekan tombol windows pada *keyboard* lalu mencari *device manager*, kemudian liat port yang terhubung akan ada keterangan NodeMCU atau bukan.
- Setelah memastikan port yang digunakan oleh Arduino Uno maka selanjutnya dilakukan pemilihan port pada Arduino IDE, dengan dengan pilihan berikut *Tools* → *Port* → ...



Gambar 9. Pemilihan Port

- Setelah memilih port yang sesuai dengan yang digunakan, Langkah terakhir adalah *upload* program.



Gambar 10. Upload Program

- Jika sudah selesai dan proses uploading berhasil maka akan muncul tampilan seperti berikut,



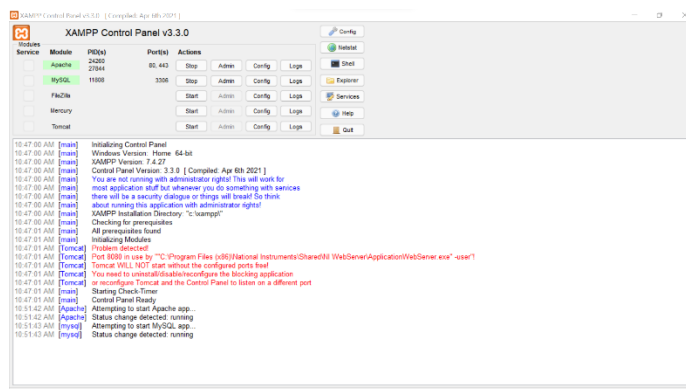
Gambar 11. Berhasil Upload

## E. Pengertian Xampp

Xampp adalah perangkat lunak berbasis *web server* yang bersifat *open source* (bebas), serta mendukung di berbagai sistem operasi, baik Windows, Linux, atau Mac OS. Xampp digunakan sebagai *standalone server* (berdiri sendiri) atau biasa disebut dengan *localhost* (Palit, 2015). Xampp juga merupakan perangkat pembantu yang menyediakan alat sebagai jembatan pembuatan sebuah program. Xampp terhubung dengan PhpMyAdmin untuk membuat sebuah platform web yang dapat menampilkan pemantauan informasi status infus. PhpMyAdmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan PhpMyAdmin, dapat membuat *data base*, tabel, meng-*insert*, menghapus dan meng-*update* data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual. Fungsi dari halaman ini adalah sebagai pengendali *data base* MySQL. Karena dengan adanya halaman ini semua hal tersebut dapat dilakukan hanya dengan meng-klik menu fungsi yang ada pada halaman PhpMyAdmin.

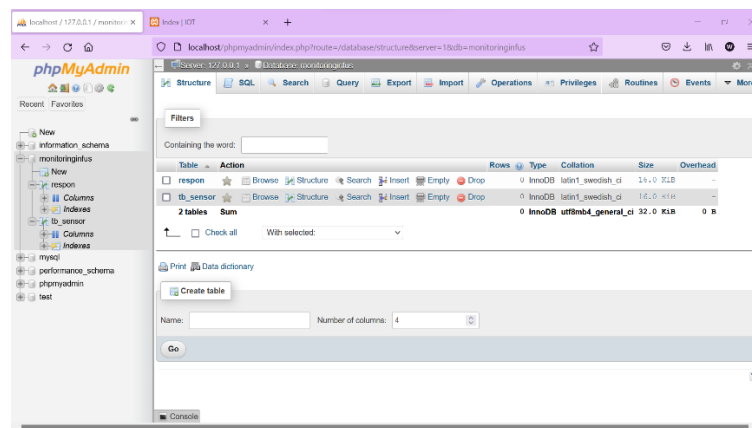
## F. Setup website untuk Menampilkan Informasi Pemantauan Infus

1. Langkah pertama adalah buka aplikasi Xampp, kemudian klik kolom *Apache* dan *MySQL*.



Gambar 12. Tampilan aplikasi XAMPP

2. Setelah itu buka *browser* dan klik di pencarian <http://localhost/phpmyadmin/>



Gambar 13. Tampilan PhpMyAdmin

3. Kemudian buat *new structure*, buat nama sesuai dengan *project* yang dibuat kemudian klik run sampai tampil *database* nya. Setelah itu, buka aplikasi *Visual Studio Code* dan ubah username sesuai dengan nama *database* pada PhpMyAdmin.

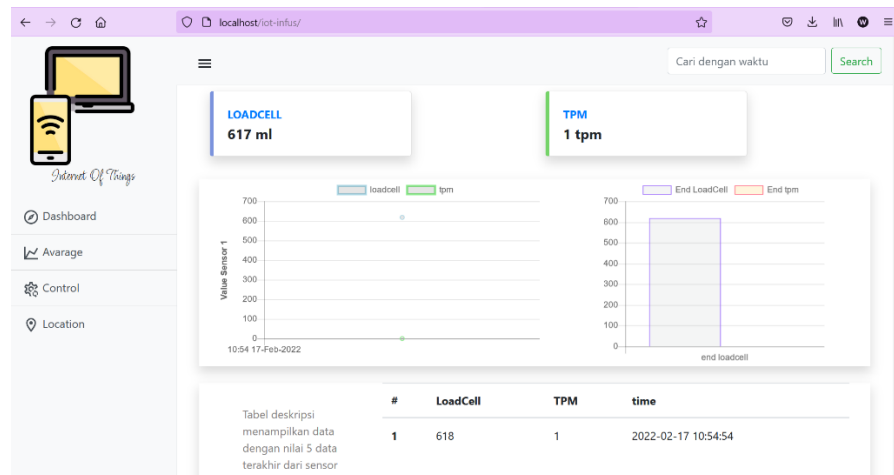
```

database.php
1 <?php
2 //year : 2020 - 2022
3
4 class database{
5
6     public static $servername = "localhost";
7     public static $database = "iot_infus";
8     public static $username = "root";
9     public static $password = "";
10
11     static public function connect(){
12         $data = mysqli_connect (
13             self::$servername,
14             self::$username,
15             self::$password,
16             self::$database);
17         // var_dump($data);
18         if($data==false){
19             echo "database not connected";
20         }
21         return $data;
22     }
23 }
24 // $static = new database;
25 // var_dump($static->connect());
26
27 >>

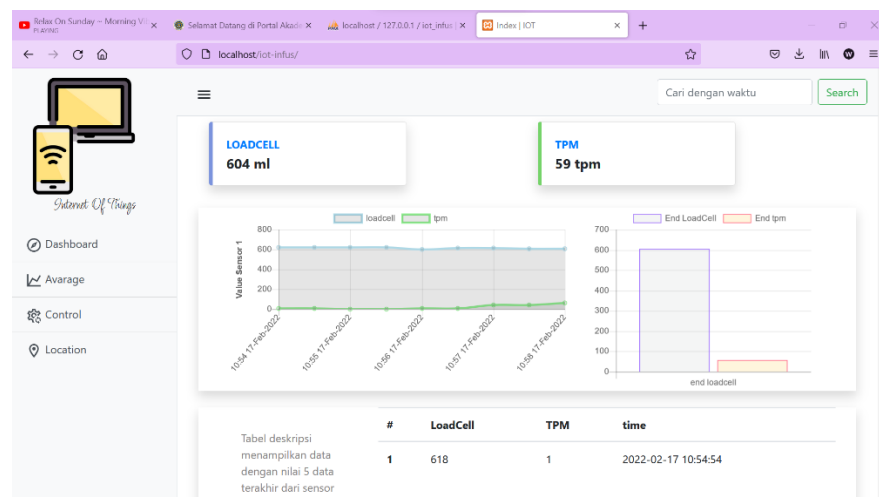
```

Gambar 14. Tampilan Visual Studio Code

- Kembali ke laman browser dan ketik alamat <https://localhost/iot-infus/> maka akan tampil tampilan informasi status infus.



Gambar 15. Tampilan Dashboard Web



Gambar 16. Tampilan Grafik Web

# Program Komputer

## 1. Arduino Uno (*Transmitter*)

```
/*
  note : transmitter
*/

#include "HX711.h"
#define DOUT A0
#define CLK A1
HX711 scale(DOUT, CLK);
float calibration_factor = 1030.10;
int GRAM;

#include <SoftwareSerial.h>
SoftwareSerial ser(5,6);

//===== TPM =====
const int sensorIR = 2; // Analog input pin that the potentiometer is attached to
int sensorValue = 0; // value read from the pot

// variables will change:
int photodiode = 0; // variable for reading the pushbutton status
int count_value = 0;
int prestate = 0;

int count_tpm;
int tpm = 0;
int trig = 0;

void tpm_sensor(){
  trig = 1;
  // int data = digitalRead(sensorIR);// not using anymore
}

void setup() {
  Serial.begin(115200);
  ser.begin(9600);

  pinMode(sensorIR, INPUT);
  scale.set_scale();
  scale.tare();
  attachInterrupt(0,tpm_sensor,FALLING);
}

void loop() {

  static uint32_t oldtime=millis();
  if ( (millis()-oldtime) > 60000) {
    oldtime = millis();
    tpm = count_tpm;
    count_tpm = 0;
  }

  /*
  * trigger sensor TPM if interrupt active
  */
  if(trig == 1){
    count_tpm ++;
    Serial.print(" count tpm = ");
    Serial.print(count_tpm);
    Serial.print("\t tpm = ");
    Serial.println(tpm);
  }
}
```

```

    }

    // reset trigger
    trig = 0;

    load_cell();
    send_serial();
}

void load_cell(){
    scale.set_scale(calibration_factor);
    GRAM = scale.get_units(), 4;
    Serial.print("Berat: ");
    Serial.print(GRAM);
    Serial.println(" ml");
}

void send_serial(){
    String datas= String(GRAM)+","+String(tpm);
    char * data = datas.c_str();
    String key[]{"ss","ff","dd"};
    ser.write("*");
    ser.write(data);
    ser.write("|");
    ser.write(data);
    ser.write("#");
    delay(1000);
}
}

```

## 2. NodeMCU ESP8266 (*Receiver*)

```

/*
 * Receiver
 */

#include <SoftwareSerial.h>
SoftwareSerial ser(D6,D5);

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>

const char *ssid = "WidyaRahayuDinata";
const char *password = "widyarahayu13";

String host = "192.168.43.103";
String url = "/iot-infus/connect.php?";

int starts = 0;
int checksum = 0;
int bypass = 0;

String tpm;
String weight;

void setup() {
    ser.begin(9600);
    Serial.begin(9600);

    // Wait for connection

```

```

    wifiConnecting();
}

void loop() {

    decoder();

    static uint32_t oldtime=millis();

    if ( (millis()-oldtime) > 30000) { //30 detik
        oldtime = millis();
        send_data();
    }

    Serial.print("tpm =>>>>>");
    Serial.println(tpm);
    Serial.print("weight =>>>>>");
    Serial.println(weight);
}

/*
 * receive data and decode with checksum
 */
void decoder(){

    String data_decode = "";
    String content = "";
    String content_check = "";

    if (ser.available()) {
        while (ser.available()) {

            char data = (char)ser.read() ;
            // Serial.print(starts);//
            if(data == '|'){
                checksum = 1;
            }
            //start
            if(data=='*')
                starts =1;
            //end
            else if(data == '#'){

                Serial.println("check");
                if(content != content_check){
                    content = "";
                    content_check = "";
                }
                data_decode = content;
                starts =0;
                checksum = 0;
                bypass =0;
                break;
            }

            else if(starts == 1){
                if(checksum == 0 )
                    content += data;

                else{
                    if(bypass == 0)
                        bypass =1;
                }
            }
        }
    }
}

```



```

        else
            content_check += data;

    }
}

/*
 * decode to partial
 */
if(data_decode != ""){
    int count_ = 0;
    String weight_staging;
    String tpm_staging;

    for (int i=0; i < data_decode.length(); i++){
        Serial.print(data_decode[i]);
        if(data_decode[i] == ','){
            count_ ++;
        }
        else{
            if(count_ == 0){
                weight_staging +=data_decode[i];
            }
            else{
                tpm_staging +=data_decode[i];
            }
        }
    }
    weight = weight_staging;
    tpm = tpm_staging;
}
Serial.println(data_decode);
delay(100);

content = "";
content_check = "";
}

void wifiConnecting(){
    //wifi
    Serial.println("");
    Serial.print("Connecting");
    //koneksi wifi
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
}

void send_data(){
    //kirim data ke server
    WiFiClient client ;
    //inisialisasi port web server 80
    const int httpPort = 80;
    if (!client.connect(host, httpPort) )
    {
        Serial.println ("gagal konek");
        return;
    }
    //kondisi pasti terkoneksi
    //kirim data sensor ke database atau web

```

```
String Link ;
HttpClient http ;

Link = "http://" + String(host)+String(url)+
"sensor1=" + String(weight) +
"&sensor2="+ String(tpm);

//eksekusi alamat link yang ada
http.begin(client,Link);
http.GET();

//baca respon setelah berhasil kirim nilai sensor
String respon = http.getString();
Serial.println(respon);
http.end();
Serial.println(Link);
```

```
}
```