

BAB III

METODOLOGI

3.1 Studi Literatur

Dalam perancangan dan pembuatan alat sistem monitoring trafo distribusi berbasis web ini dibutuhkan sumber-sumber referensi sebagai bahan acuan dan pertimbangan. Sumber referensi didapatkan dari sumber langsung dan tak langsung. Sumber langsung didapat dari hasil diskusi atau konsultasi dengan dosen, sedangkan sumber tak langsung didapat dari tulisan laporan penelitian-penelitian yang dilakukan sebelumnya, buku, internet serta referensi-referensi lain yang berkaitan dengan perancangan dan pembuatan alat.

3.2 Perancangan Sistem

Alat monitoring trafo distribusi online adalah alat untuk memonitor Arus, tegangan, suhu dan beban daya trafo distribusi yang pembacaannya secara real time melalui sebuah web. Didalam alat ini terdiri dari bagian bagian yaitu *Current Transformer*(CT) yang berfungsi untuk mengukur arus yang mengalir. *Current Transformer* tersebut tidak bisa tersambung langsung ke STM 32. Untuk menghubungkan CT ke STM32 menggunakan converter agar dapat terbaca oleh STM32. Konverter dalam hal ini yaitu mengkonversi arus menjadi tegangan dan level tegangan pada ADC yaitu 3,3 V. Untuk dsain rangkaian converter adalah

Misalnya nilai CT : 200/5 A

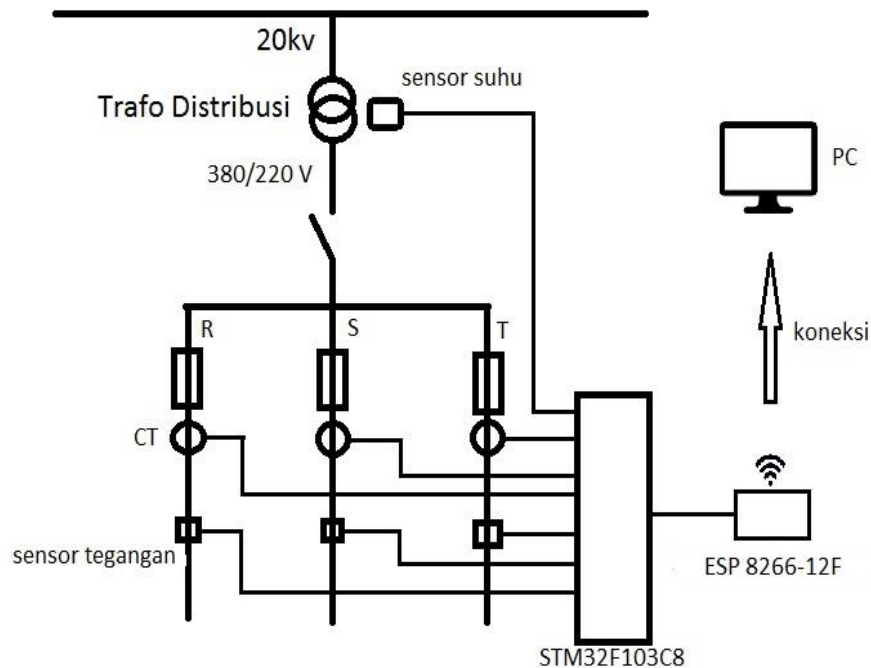
$$\begin{aligned} 200A \text{ RMS} &= \sqrt{2} * 200 \\ &= 1,414 * 200 \\ &= 282,8 \text{ VPP} \end{aligned}$$

$$\begin{aligned} \text{Turn} &= 200/5 \\ &= 40 \end{aligned}$$

$$\begin{aligned} I (\text{sens}) &= I (\text{peak}) / 40 \\ &= 282,8 / 40 \\ &= 7,07 \text{ A} \end{aligned}$$

$$\begin{aligned} R (\text{burden}) &= \frac{3,3/2}{7,07} \\ &= 0,2334\Omega \\ &= 0,22\Omega \end{aligned}$$

Sensor tegangan yaitu untuk mengukur tegangan antara phasa – nol dan phasa – phasa. Sensor suhu yaitu untuk mengukur kelembapan atau suhu pada trafo. Semua alat itu terhubung ke sebuah mikrokontroler STM32



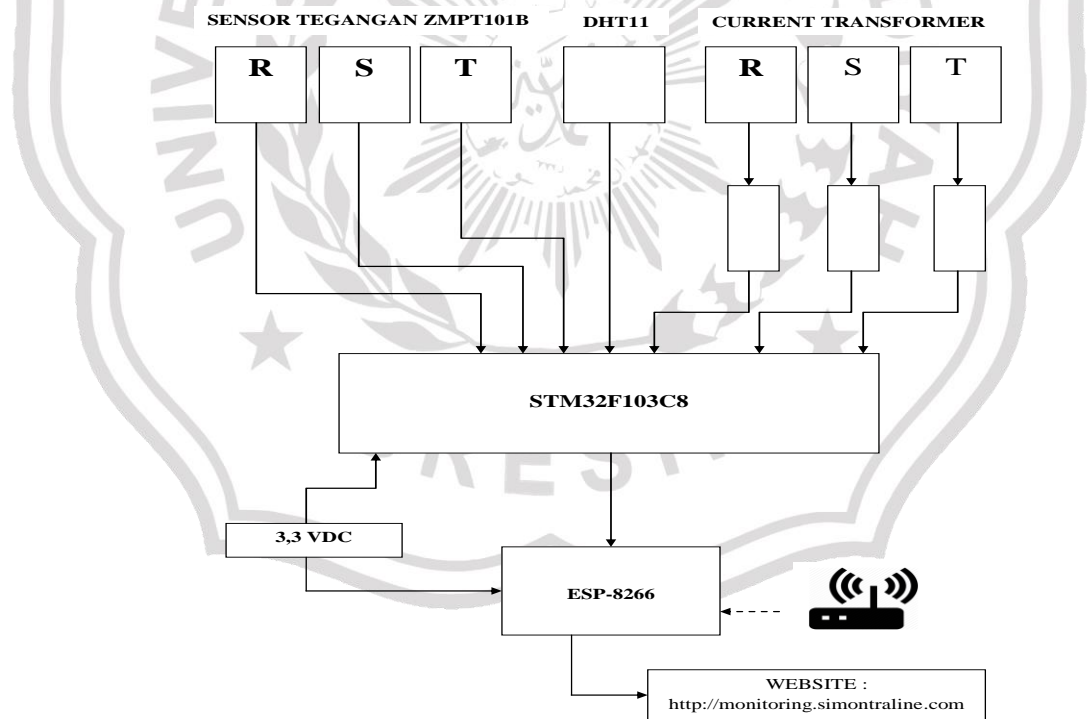
Gambar 3.1 Skema global sistem monitoring Trafo online

Didalam mikrokontroler tersebut olah nilai arus dan tegangan yang didapat dari CT dan sensor tegangan menjadi sebuah daya beban trafo. Nilai arus, tegangan, daya beban trafo dan suhu ditampilkan di LCD. Selanjutnya data dari pengukuran tersebut dikoneksikan ke sebuah modul Wifi yaitu ESP 8266 yang tersambung dengan wi-fi router. Setelah terkoneksi ke web maka diketahui hasil pengukuran tersebut di sebuah web yaitu <http://monitoring.simontraline.com>. Data yang dikirim ke web berupa script dan proses pembacaannya per 1 menit. Pembacaan data pada web secara real time, apabila ingin mengambil data waktu sebelumnya kita gunakan USB

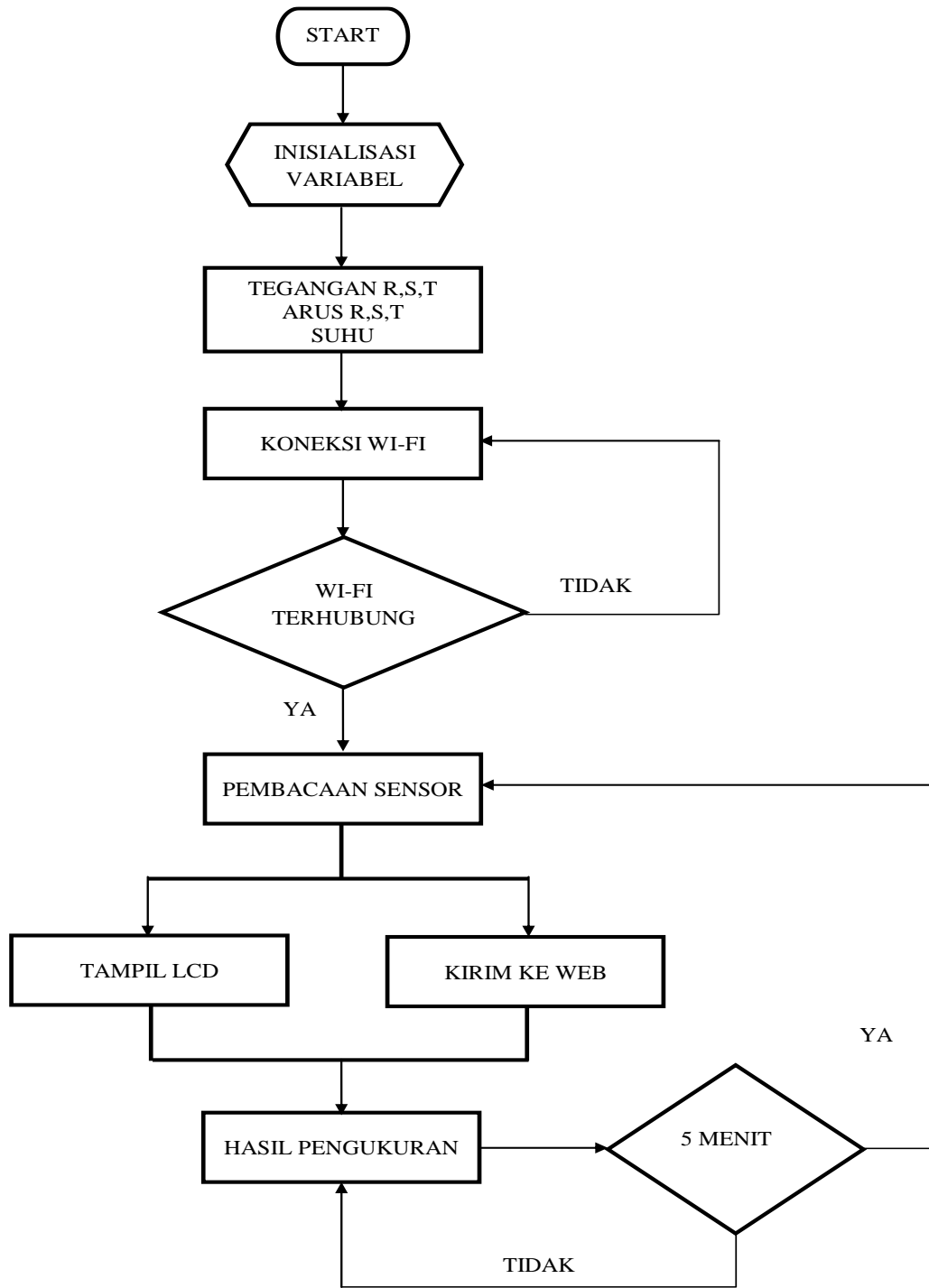
serial. Alat sistem monitoring trafo distribusi berbasis online menggunakan Mikrokontroler STM32 dirancang dengan konstruksi secara garis besar terdiri dari dua perangkat utama yaitu:

1. Perangkat keras (hardware), yaitu berupa rangkaian mikrokontroler STM32 dan beberapa sensor.
2. Perangkat lunak (software), yaitu alur program yang dibuat untuk menjalankan sistem sensor.[10]

Gambar dibawah ini menunjukkan diagram blok sistem secara umum atau keseluruhan dari alat sistem monitoring trafo distribusi berbasis mikrokontroler STM32.



Gambar 3.2 Diagram blok sistem monitoring Trafo Distribusi online



Gambar 3.3 Flow Chart pengukuran trafo online

3.3 Perancangan alat

Perancangan alat secara garis besar terdapat 3 komponen utama.

- Pertama, rangkaian monitoring yang terdiri dari STM32F103, LCD, sensor tegangan, sensor arus (*Current Transformer*), modul *WIFI*, sensor suhu dan sumber dc untuk sumber tegangannya alat.
- Kedua, website sebagai alat untuk menyimpan data yang dihasilkan oleh rangkaian control setiap satu menit, dan sebagai alat interfacing mobile yang bisa diakses ditempat yang ditentukan.
- Ketiga, gardu distribusi adalah sumber tegangan atau arus sebagai sumber yang akan di monitoring keluarannya dan kondisinya.

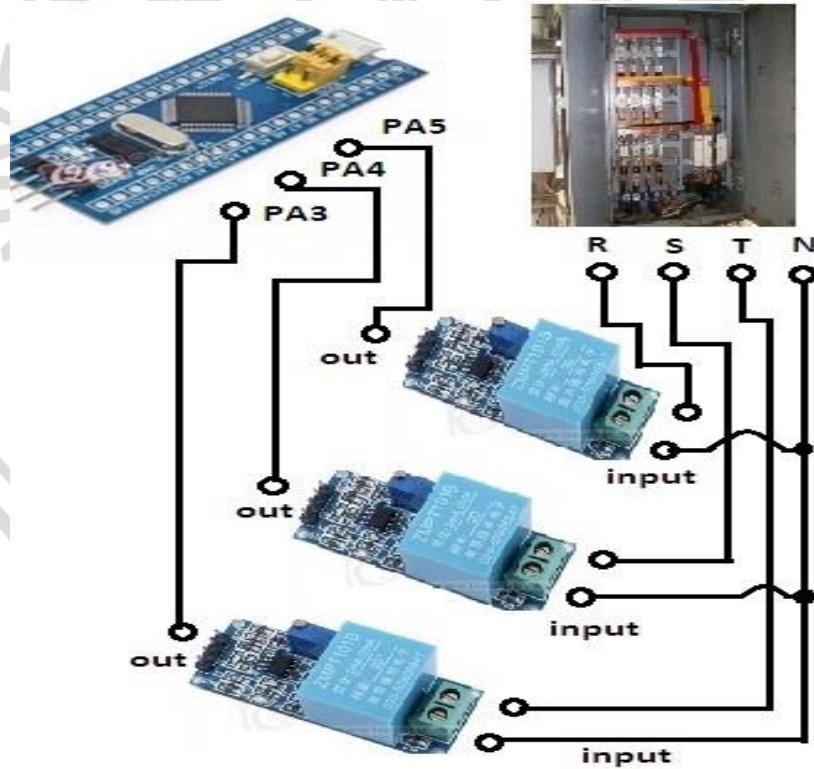
3.3.1 Perancangan elektronika

Perancangan elektronika dalam pembuatan alat monitoring Trafo Distribusi online ini adalah rancangan satu buah Trafo Distribusi sebagai obyek yang akan diukur kondisinya dan satu buah rangkaian alat monitoring sebagai alat ukur serta alat komunikasi pada website yang ditentukan.

Fungsi dari monitoring tersebut untuk mengetahui kondisi Trafo Distribusi mencakup beban dan batas beban maksimal daya Trafo Distribusi.

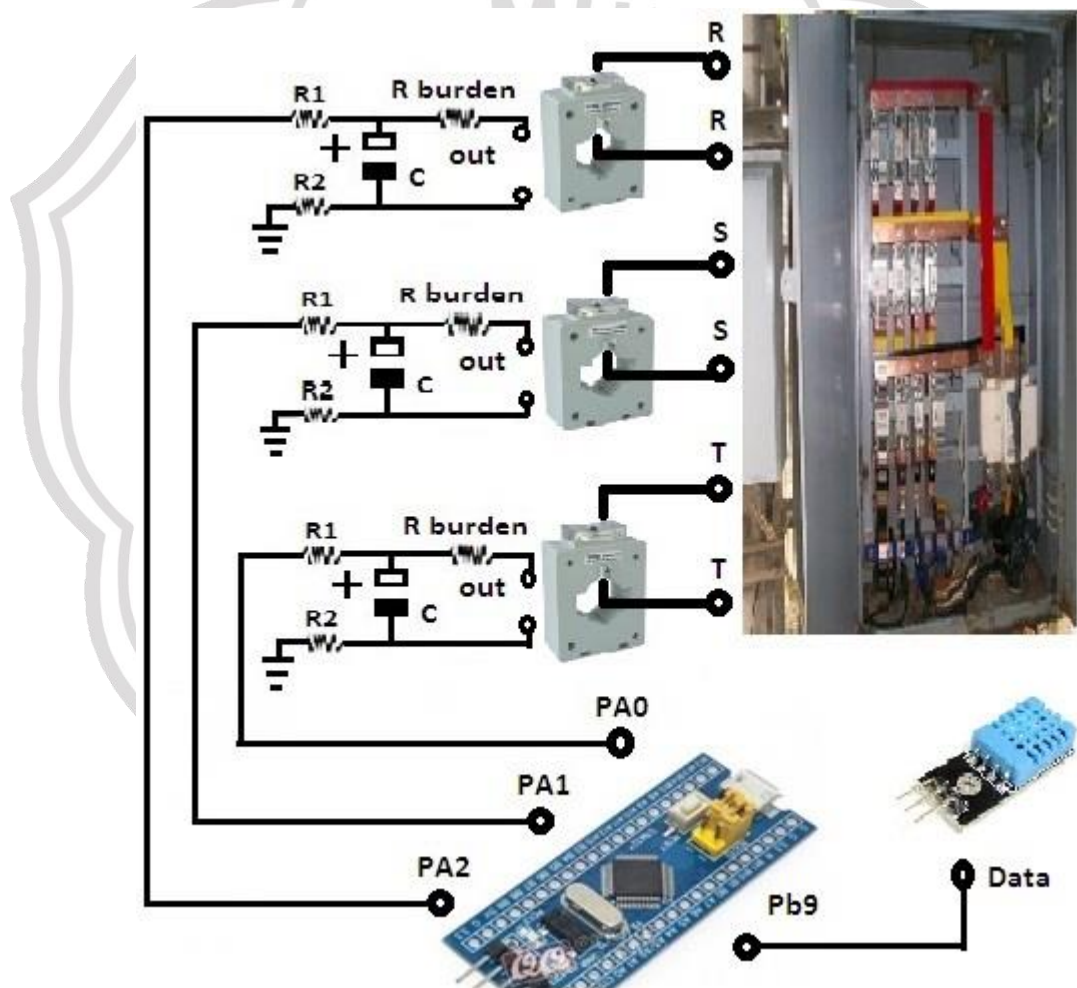
Pada tahap ini alat yang sebelumnya sudah dirangkai akan dipasang satu buah sensor tegangan sebagai alat ukur tegangan, satu buah sensor arus (*Current Transformer*) sebagai alat ukur arus, modul *WIFI* sebagai gerbang koneksi alat ke jaringan dan satu buah STM32F103.

Dalam perancangan sensor tegangan, sumber tegangan disambungkan pada input sensor tegangan, karena modul tegangan ini terdapat fitur kalibrasi manual maka tidak perlu di tambahkan pembagi tegangan, disetiap output / keluaran dari sensor tegangan langsung disambungkan pada pin A5 A4 dan A3, seperti gambar 3.4.



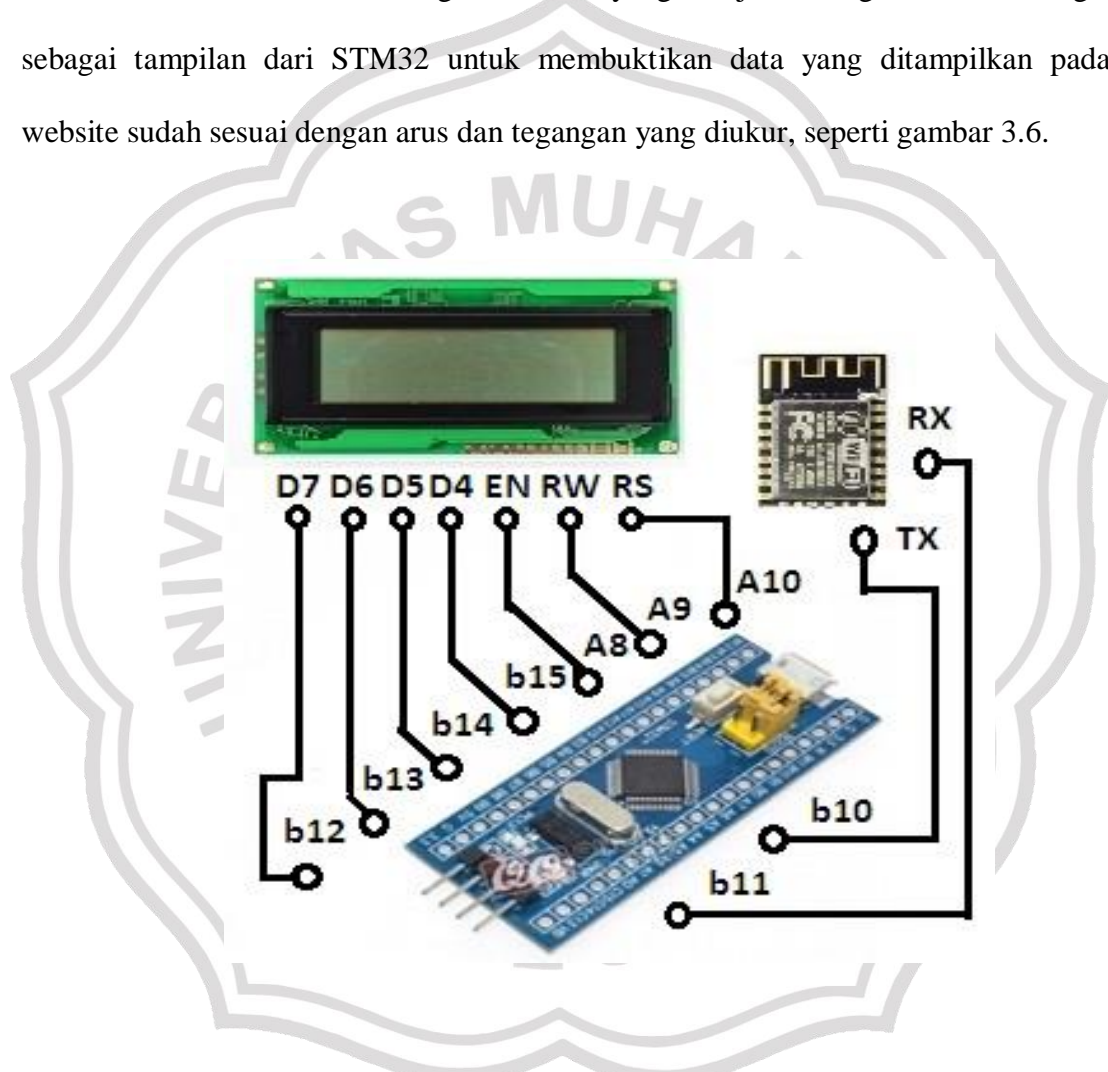
Gambar 3.4 Rangkaian sensor tegangan

Sedangkan sensor arus (*Current Transformer*) tengahnya dilewati kabel phase dari trafo yang diukur, output / keluaran sensor arus disambungkan pada rangkaian converter arus ke tegangan sehingga tegangan yang masuk sesuai dengan tegangan adc pada stm32. Untuk sensor suhu, pin b9 dihubungkan pada pin data dht11 sebagai komunikasi *single-wire bi-directional* (kabel tunggal dua arah). seperti gambar 3.5.



Gambar 3.5 Rangkaian sensor arus

Pada rangkaian modul wifi (ESP8266) dan tampilan, pin rx dan tx disambungkan STM32 pada pin pb11 dan pb10 untuk komunikasi serial, dan lcd disambungkan pada pin telah ditentukan. Modul wifi berfungsi sebagai gerbang komunikasi antara STM32 dengan website yang dituju, sedangkan lcd berfungsi sebagai tampilan dari STM32 untuk membuktikan data yang ditampilkan pada website sudah sesuai dengan arus dan tegangan yang diukur, seperti gambar 3.6.



Gambar 3.6 Modul wifi dan lcd

3.3.2 Perancangan koneksi website

Hasil pengukuran yang didapat pada rancangan elektronika dikirimkan melalui esp8266 ke website yang ditentukan, esp8266 disambungkan pada wi-fi router sehingga didapatkan jalur akses internet yang menghubungkan antara stm32 dengan website yang dituju, pengiriman data dilakukan setiap 1 menit yang tersimpan pada website dan data pengukuran dapat diunduh berupa file Microsoft Exel.

Berikut program Esp8266 menggunakan Arduino-IDE

```
#include <Adafruit_ESP8266.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
const char *ssid = "bismillah"; //ENTER YOUR WIFI SETTINGS
const char *password = "bismillah";
const char tail = '\n', header = '$';
String payload, url;
char cmd_idx;

#define cmd_erase 'E'

#define cmd_write 'S'

#define cmd_read 'G'
```

```

#define cmd_con_wifi 'W'

#define cmd_req 'R'

#define cmd_status 'F'

#define cmd_host 'H'

#define cmd_read_con 'O'

void send_resp(bool statusFlag)
{ if(statusFlag)
Serial.println("$F,1");
else
Serial.println("$F,0");
}

bool write_to_server(String db, int len, String param, String val)
{ HTTPClient http; //Declare object of class HTTPClient
String postData, getData;
bool statusFlag = false;
if(WiFi.status() == WL_CONNECTED)
{ postData = "dbtable=";

postData += db;

postData += "&length=";

postData += String(len);

```

```

postData += "&parameter=";

postData += param;

postData += "&value=";

postData += val;

http.begin("http://monitoring.simontraline.com/updatedatakwh.php");

http.addHeader("Content-type", "application/x-www-form-urlencoded");

int httpCode = http.POST(postData); //Send the request
    String payload = http.getString(); //Get the response payload
http.end(); //Close connection
if(payload.indexOf("OK") > 0)
    statusFlag = true;
    }
return statusFlag;
}

bool write_to_database()
{ char separator = ',';
byte start_idx, end_idx;

String param = "", nilai = "";

start_idx = payload.indexOf(separator);

end_idx = payload.indexOf(separator, start_idx + 1);

String db = payload.substring(start_idx + 1, end_idx);

```

```

start_idx = end_idx;

end_idx = payload.indexOf(separator, start_idx + 1);

int len = payload.substring(start_idx + 1, end_idx).toInt();

for(int i=0; i<len; i++)

    {start_idx = end_idx;

    end_idx = payload.indexOf(separator, start_idx + 1);

    param += payload.substring(start_idx + 1, end_idx);

    if(i + 1 < len)

        { separator = '!';

        param += ",";

        }

    else

        separator = tail;

        start_idx = end_idx;

        end_idx = payload.indexOf(separator, start_idx + 1);

        nilai += payload.substring(start_idx + 1, end_idx);

        if(i + 1 < len)

            nilai += ",";

        }

    if(!write_to_server(db, len, param, nilai))

return false;

```

```

return true;

}

bool wait_data_sensor(unsigned long timeout)

{ unsigned long last_millis;

bool header_flag;

Serial.println("$R;");

last_millis = millis();

while(millis() - last_millis < timeout)

{ if(Serial.available())

{ char c = Serial.read();

if(c == header)

{ last_millis = millis();

header_flag = true;

payload = "";

}

else if(c == tail && header_flag == true)

{ payload += c;

cmd_idx = payload[0];

return true;

}

else if(header_flag == true)

```



```

payload += c;

    } }

return false;

}

void setup() {

delay(1000);

Serial.begin(115200);

WiFi.mode(WIFI_OFF delay(1000);

WiFi.mode(WIFI_STA WiFi.begin(ssid, password);

Serial.println("");

Serial.print("Connecting");

while (WiFi.status() != WL_CONNECTED)

{ delay(500);

Serial.print(".");

}

Serial.println("");

Serial.print("Connected to ");

Serial.println(ssid);

Serial.print("IP address: ");

Serial.println(WiFi.localIP()); //IP address assigned to your ESP

}

```

```
void loop()
{
  if(wait_data_sensor(1000))
  {
    switch(cmd_idx)
    {
      case cmd_write:
        send_resp(write_to_database());
      break;
      case cmd_erase:
      break;
      case cmd_con_wifi:
      break;
      case cmd_read:
      break;
      case cmd_host:
      break;
      case cmd_read_con:
      break;
      default:
      break;
    }
  }
}
```