

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Alat Dan Bahan

##### 3.1.1 Alat

Berikut adalah alat yang akan digunakan dalam pembuatan modul, perhatikan Tabel 3.1 berikut ini.

**Tabel 3.1** Daftar Alat

No	Alat	Jumlah
1	Toolset	1
2	Laptop	1
3	Water heater	1
4	Mini drill	1
5	Setrika	1
6	Solder	1
7	Spidol permanen	Seperlunya
8	Gunting	1

##### 3.1.2 Bahan

Berikut adalah bahan yang akan digunakan dalam pembuatan modul, perhatikan Tabel 3.2 berikut ini.

**Tabel 3.2** Daftar Bahan

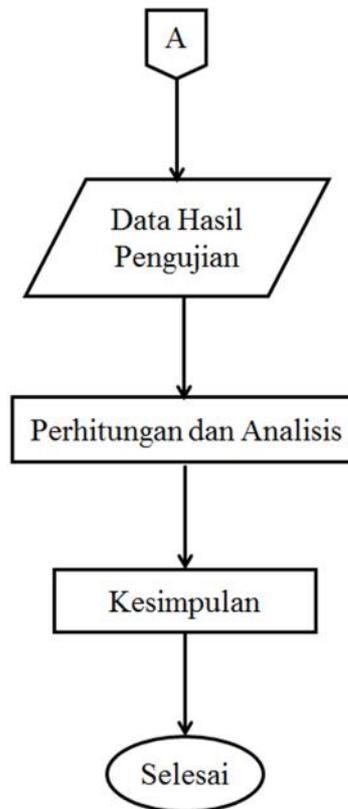
No	Bahan	Jumlah
1	Air flow sensor	1
2	Oscilator kristal	1
3	ATmega328	1
4	Resistor	Seperlunya
5	Kapasitor	Seperlunya
6	Dioda	Seperlunya
7	Transformator	1
8	Kabel	Seperlunya
9	Push Button	Seperlunya

### 3.2 Alir Penelitian

Alir penelitian berfungsi untuk menjelaskan tentang proses atau langkah-langkah yang dilakukan pada penelitian ini. Gambar 3.1 menunjukkan diagram alir dari penelitian.



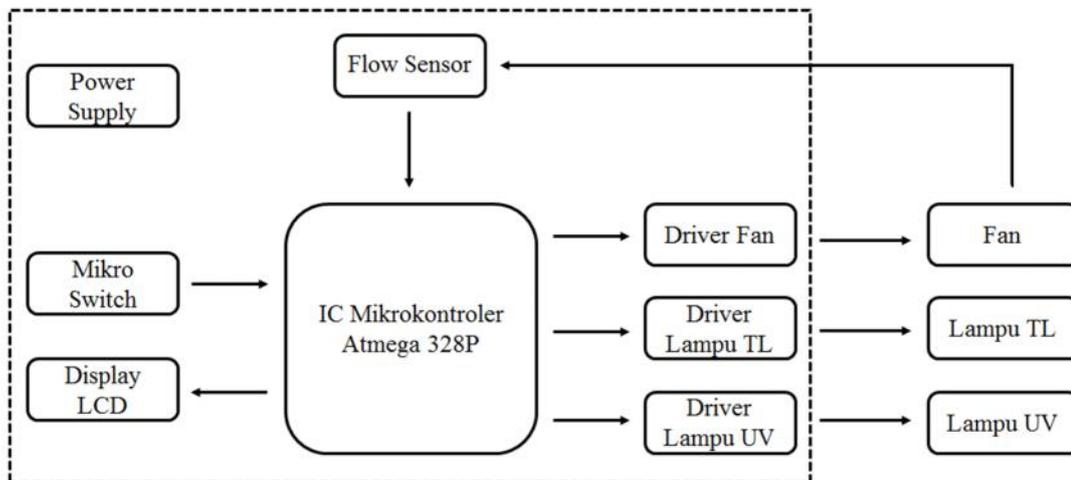
**Gambar 3.1** Alir Penelitian



**Gambar 3.1** Alir Penelitian (Lanjutan)

### 3.3 Implementasi Perangkat Keras

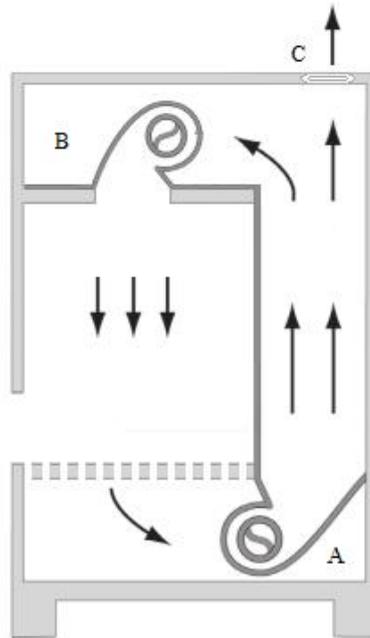
Perangkat keras merupakan komponen komponen yang dapat terlihat dengan kasat mata serta memiliki bentuk fisik. Dalam perancangan perangkat keras, dibutuhkanlah suatu blok diagram untuk mempermudah perancangan perangkat keras. Dari blok diagram, dapat diketahui rangkaian dan perangkat keras apa saja yang akan digunakan di dalam sistem alat. Adapun rangkaian yang digunakan dalam blok diagram ini terdiri dari rangkaian catu daya, rangkaian, sensor, rangkaian microcontroller ATmega 328P, rangkaian *driver*, serta rangkaian LCD. Gambar 3.2 menunjukkan blok diagram dari alat.



**Gambar 3.2** Diagram Blok

Pada Gambar 3.2 tegangan listrik dari PLN sebesar 220V AC akan disearahkan oleh rangkaian *power supply* yang nantinya akan mensuplai tegangan DC ke seluruh rangkaian. Setelah itu, ketika tombol *micro switch* ditekan maka *timer 1* pada *microcontroller* akan berjalan dan memberikan *delay*. Ketika *delay* habis, maka *timer 2* akan bekerja dan *microcontroller* akan memberikan logika pada *driver UV* sehingga lampu UV akan bekerja selama 30 menit.

Setelah *timer 2* habis, maka lampu UV akan mati. Setelah itu, *microcontroller* akan memberikan logika pada *driver* lampu TL dan *driver fan* secara bersamaan sehingga lampu TL sebagai penerangan akan menyala dan pada saat yang sama pula *fan* akan bekerja. Udara yang dihasilkan oleh *fan* akan membentuk aliran udara yang bergerak di dalam *laminar*. Gambar 3.3 menunjukkan alir dari aliran udara yang bergerak di dalam *laminar*.



**Gambar 3.3** Aliran Udara Di Dalam *Laminar*

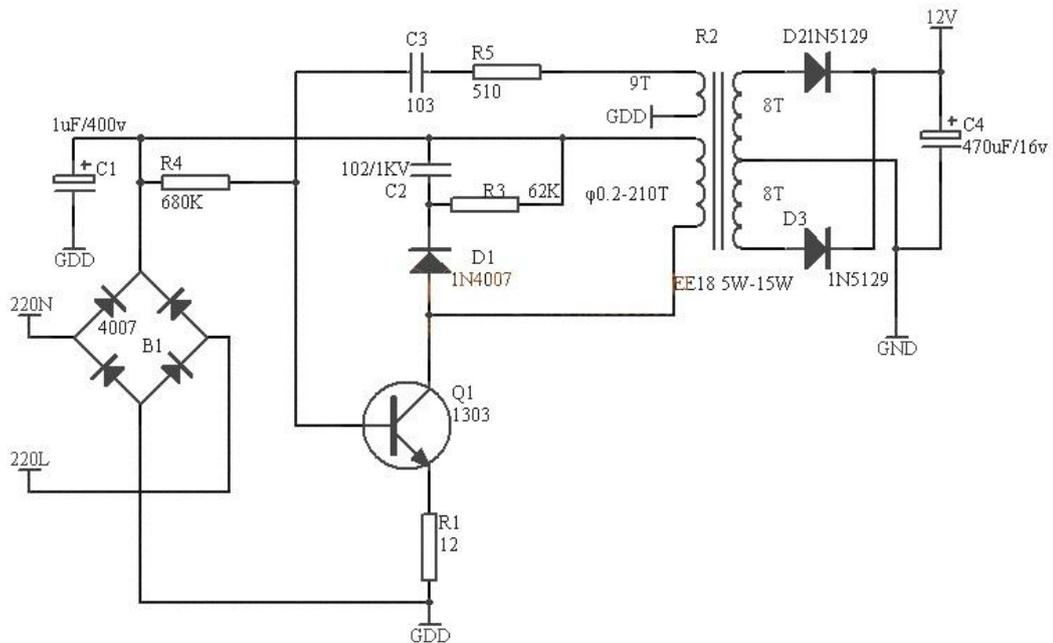
Pada Gambar 3.3 aliran udara dihasilkan oleh *fan A* dan *fan B*, *fan A* sebagai *fan* penghisap udara sedangkan *fan B* merupakan *fan* penghembus udara. Aliran udara di dalam *laminar* tersebut akan dideteksi oleh *flow sensor*. Besaran aliran udara akan diubah menjadi tegangan oleh sensor, sehingga tegangan *output* sensor akan masuk ke pin ADC pada *microcontroller*. *Microcontroller* tersebut akan mengubah data analog dari sensor menjadi digital sehingga kecepatan aliran udara dapat ditampilkan pada LCD.

Untuk penjelasan tentang perancangan perangkat keras yang digunakan dalam sistem ini, dapat dijelaskan sebagai berikut.

### 3.3.1 Catu Daya

Catu daya atau *power supply* merupakan sebuah perangkat yang berguna sebagai sumber listrik untuk perangkat lainnya. Tanpa adanya masukan daya, maka perangkat tidak dapat berfungsi. *Power supply* yang digunakan pada perancangan

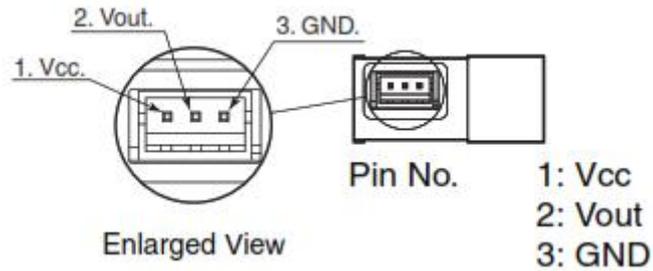
alat ini adalah *power supply switching* 12V 5A. Gambar 3.4 menunjukkan rangkaian dari *power supply switching* 12 volt.



**Gambar 3.4** Rangkaian *Power Supply Switching* 12 Volt

### 3.3.2 Flow Sensor OMRON D6F-W04A1

*Flow* sensor yang diproduksi oleh OMRON dengan seri D6F-W04A1 merupakan sensor yang digunakan pada sistem untuk mengukur laju aliran udara yang mengalir di dalam kabinet. Prinsip kerja sensor ini udara akan masuk ke dalam sensor dan akan melewati sebuah *chip* yang dinamakan *micro electro mechanical system (MEMS) flow chip*. *Chip* ini akan mengubah aliran udara tersebut menjadi sinyal listrik analog, kemudian sinyal listrik yang sangat kecil tersebut akan dikuatkan sehingga sinyal *output* sensor yang keluar menjadi besar. Gambar 3.5 menunjukkan konfigurasi pin dari sensor OMRON D6F-W04A1.



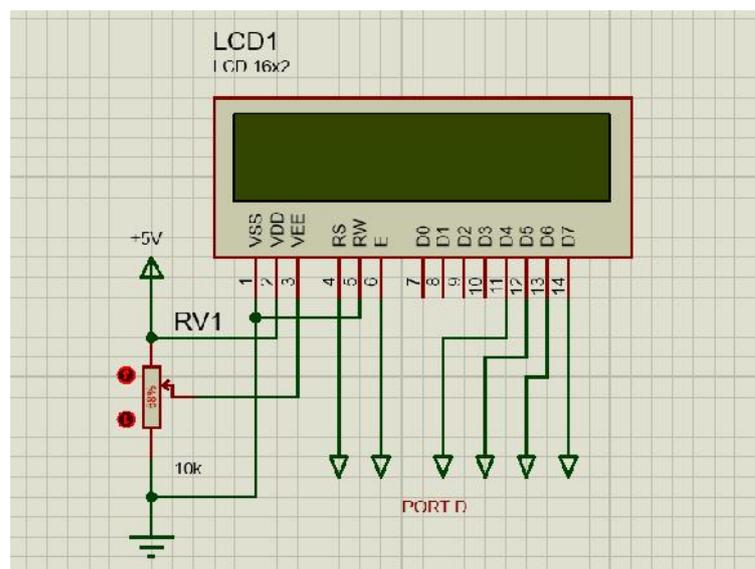
**Gambar 3.5** Konfigurasi Pin Sensor OMRON D6F-W04A1

Berdasarkan pada Gambar 3.5 dapat dijelaskan konfigurasi pin dari sensor sebagai berikut.

- Pin VCC dihubungkan pada *power supply* 12V DC
- Pin Vout dihubungkan pada pin PC0 pada *microcontroller* ATmega 328P
- Pin GND dihubungkan pada pin GND pada *microcontroller* ATmega 328P

### 3.3.3 LCD 16x2

Implementasi dari rangkaian penampil pembacaan laju aliran udara di dalam *laminar* dengan menggunakan LCD 16x2. Gambar 3.6 menunjukkan rangkaian dari LCD 16x2.



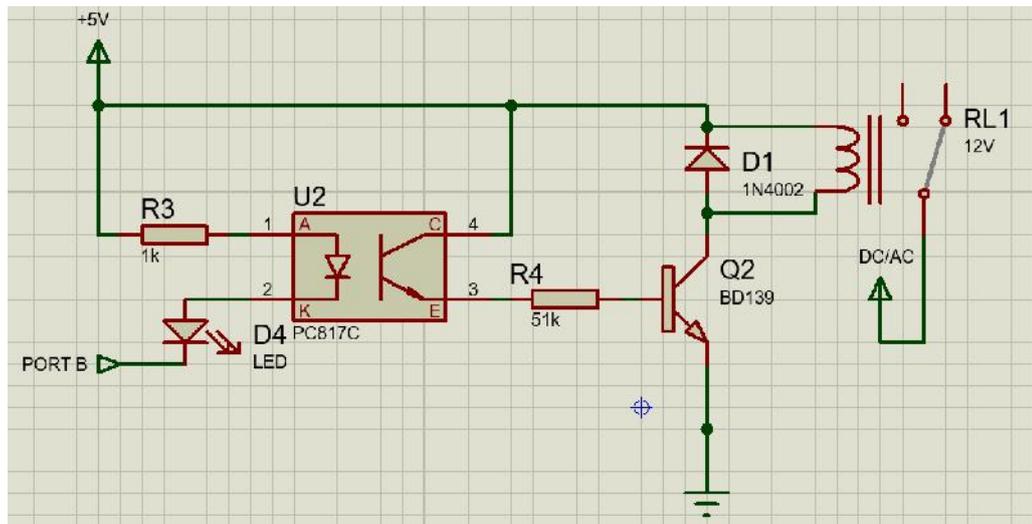
**Gambar 3.6** Rangkaian LCD 16x2

Berdasarkan pada Gambar 3.6 dapat dijelaskan konfigurasi pin pada LCD 16x2 sebagai berikut.

- a. Pin RS dihubungkan pada pin PD2 pada *microcontroller* ATmega 328P
- b. Pin E dihubungkan pada pin PD3 pada *microcontroller* ATmega 328P
- c. Pin D4 dihubungkan pada pin PD4 pada *microcontroller* ATmega 328P
- d. Pin D5 dihubungkan pada pin PD5 pada *microcontroller* ATmega 328P
- e. Pin D6 dihubungkan pada pin PD6 pada *microcontroller* ATmega 328P
- f. Pin D7 dihubungkan pada pin PD7 pada *microcontroller* ATmega 328P
- g. Pin VSS dihubungkan pada pin GND pada *microcontroller* ATmega 328P
- h. Pin VDD dihubungkan pada pin 5v pada *microcontroller* ATmega 328P
- i. Pin VEE dihubungkan pada *output* potensio, pin ini berfungsi sebagai pengatur kecerahan pada LCD.

#### 3.3.4 *Driver Fan, Lampu UV, dan Lampu TL*

Untuk menyalakan *fan* DC, lampu UV, dan lampu TL pada umumnya dibutuhkan sebuah *driver* karena sumber yang digunakan untuk menyuplai komponen tersebut berbeda dengan sumber yang digunakan untuk menyuplai *microcontroller*. *Driver* ini berfungsi sebagai pemisah untuk mencegah terjadinya penarikan arus yang berlebih dari *microcontroller*, sehingga menjaga agar *microcontroller* tidak akan cepat rusak. Gambar 3.7 menunjukkan rangkaian dari *driver fan*, lampu UV, dan lampu TL.

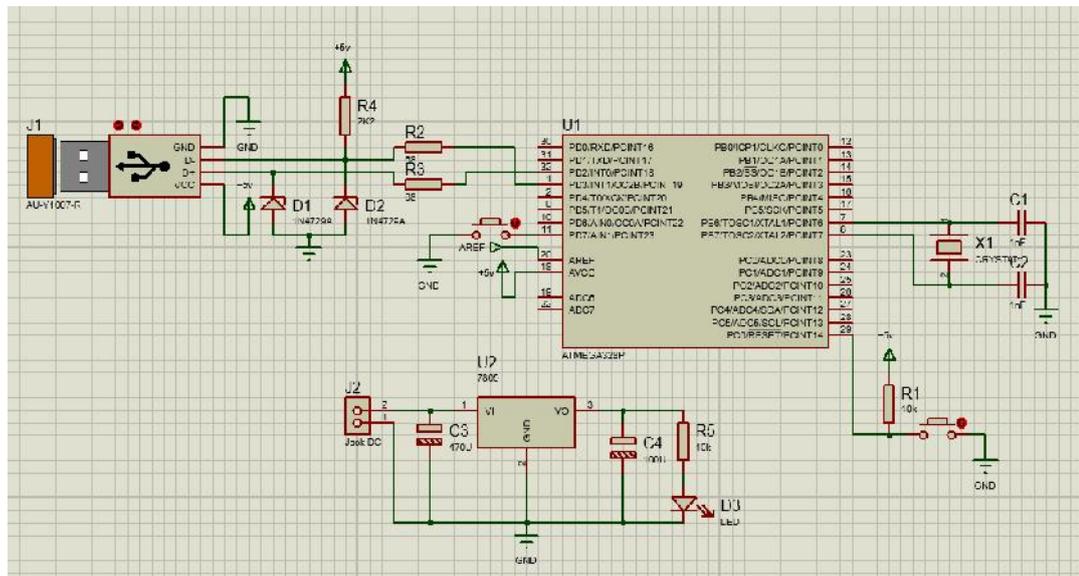


**Gambar 3.7** Rangkaian *Driver Motor, Driver Lampu UV, dan Lampu TL*

Rangkaian pada Gambar 3.7 menggunakan 5v DC sebagai suplainya. Ketika pin 2 pada *optocoupler* mendapat logika *low* dari *microcontroller*, maka *optocoupler* akan bekerja. Transistor yang terdapat pada *optocoupler* akan saturasi, sehingga akan mentrigger basis transistor BD139 dan mengakibatkan transistor tersebut saturasi. Saat transistor saturasi, kaki kolektor dan emitor terhubung sehingga relay mendapatkan GND dan *relay* menyala. Hal ini menjadikan *relay* yang berada pada *normally close* akan menjadi *normally open*.

### 3.3.5 *Microcontroller* ATmega 328P

*Microcontroller* merupakan komponen penting dalam sistem. *Microcontroller* sendiri berfungsi sebagai pengontrol rangkaian elektronik dan dapat menyimpan program di dalamnya. Gambar 3.8 menunjukkan rangkaian *microcontroller* ATmega 328P.



**Gambar 3.8** Rangkaian *Microcontroller* ATmega 328P

Rangkaian pada Gambar 3.8 menggunakan catu daya 5 volt sebagai sumber tegangan ke IC *microcontroller*. IC catu daya atau regulator yang digunakan adalah IC seri LM7805 yang mampu mensuplai arus sekitar 1 A. Untuk rangkaian osilator eksternal digunakan *crystal* sebesar 16 MHz. Osilator disini berfungsi sebagai pembangkit *clock* atau detak pada *microcontroller*. Sedangkan untuk rangkaian antarmuka ISP tidak dipasangkan pada rangkaian tersebut, karena proses mengisi atau men-*download* program dapat dilakukan melalui *port* USB secara langsung. *Port* USB terhubung langsung ke PD2 dan PD3 pada *microcontroller* ATmega 328P. Perlu diketahui pin D+ USB harus terhubung ke pin INT0 (*interrupt*) yaitu pin PD3. Sedangkan pin D- USB dapat bebas dihubungkan ke pin lainnya yang berada pada *PORT D*.

*Push button* yang terhubung ke PD7 digunakan ketika ingin mengisi atau mendownload program. Jika *switch* tersebut ditekan bersamaan dengan *switch reset* maka *bootloader* USBasploader akan aktif dan *microcontroller* akan masuk ke

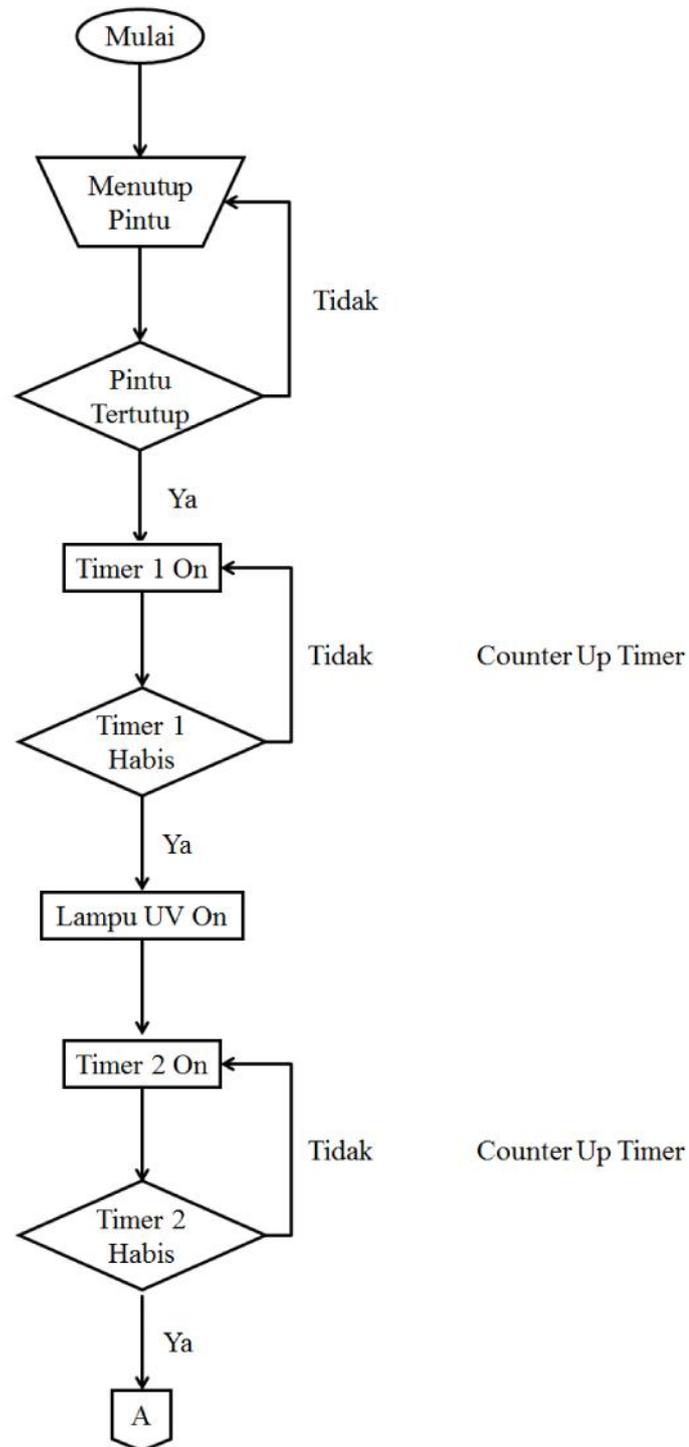
mode program. Namun jika *switch* ini tidak ditekan ketika *reset* maka *microcontroller* akan mengeksekusi program yang telah diisikan ke dalam memorinya. Konfigurasi hubungan pin D+ D- dan *switch* mode program akan disesuaikan pada *source code firmware* USBasploader.

IC ATmega 328P yang merupakan IC *microcontroller* pada minimum sistem arduino yang dibuat harus diisi dengan *firmware bootloader* USBasploader terlebih dahulu. Setelah *firmware* USBasploader diisi pada *microcontroller*, minimum sistem dapat digunakan dan diprogram agar dapat digunakan.

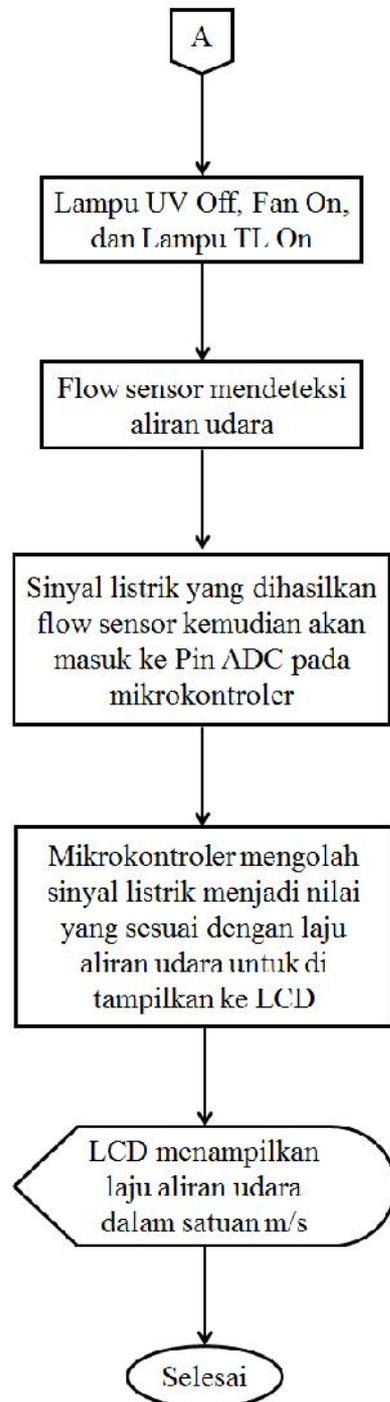
### **3.4 Implementasi Perangkat Lunak**

Perangkat lunak atau yang biasa dikenal dengan *software* adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer. Data elektronik tersebut dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Dengan begitu perangkat lunak hanya dapat dilihat dengan hasil kerjanya saja dan tidak bisa dilihat secara fisik karena tidak memiliki bentuk fisik. Perangkat lunak ini yang akan digunakan untuk mengontrol kerja perangkat keras di dalam sistem alat.

Arduino IDE merupakan aplikasi yang digunakan dalam pembuatan perangkat lunak dari alat. arduino IDE menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Dalam pembuatan perangkat lunak dibutuhkanlah suatu diagram alir, agar dapat diketahui sistem kerja alat yang diinginkan berjalan seperti apa. Gambar 3.9 menunjukkan diagram alir dari sistem alat.



**Gambar 3.9** Diagram Alir



**Gambar 3.9** Diagram Alir (Lanjutan)

Berdasarkan Gambar 3.9 saat *microswitch* pada posisi ON maka alat akan menyala. Apabila pintu masih terbuka, maka alat tidak akan beroperasi. Pada saat pintu ditutup, *timer 1* akan mulai menghitung. Apabila *timer 1* habis, maka lampu UV dan *timer 2* akan menyala selama 30 menit. Kemudian *timer* akan menghitung dan ditampilkan pada LCD. Ketika *timer 2* habis, lampu UV akan mati serta lampu TL dan *fan* akan menyala secara bersamaan.

Aliran udara yang dihasilkan oleh *fan* akan di deteksi oleh *flow sensor* dan kemudian akan diolah oleh *microcontroller* untuk di tampilkan pada *display*. Sistem lampu TL dan *fan* terus berjalan sampai *switch* pada posisi *off*.

*Laminar flow biological safety cabinet class II type B3* dilengkapi dengan tampilan LCD berbasis *microcontroller* ATmega 328P menggunakan bahasa pemrograman Arduino IDE dalam pemrogramannya. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam pemrograman dari bahasa aslinya.

Program Arduino IDE adalah program utama yang digunakan untuk menampilkan laju aliran udara di dalam *laminar* pada layar LCD, *flow sensor* OMRON D6F-W04A1 untuk membaca aliran udara dan diubahnya menjadi sinyal listrik yang akan diprogram oleh *microcontroller* untuk menjadi nilai laju aliran udara, menyalakan *driver* sehingga *fan*, lampu TL, dan lampu UV menyala. Penyusunan program dibagi menjadi beberapa bagian yaitu program *header*, program *setup*, program *loop*, program *timer* ISR, dan program baca sensor.

### 3.4.1 Listing Program Header

*Listing* program *header* ini berisi perintah pemanggilan *library* komponen.

*Listing* 3.1 menunjukkan *listing* program dari program *header*.

```
// library timer 1
#include <TimerOne.h>
// library lcd
#include <LiquidCrystal.h>
// konfigurasi lcd
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

// konfigurasi i/o
#define ledpin      13
#define sensorpin  A0
#define uvpin      8
#define tlpin      9
#define fan1pin    10
#define fan2pin    11
#define tombolpin  12
```

**Listing 3.1** Listing Program Header

Pada *Listing* 3.1 dapat dijelaskan bahwa kode “*#include*” digunakan untuk memanggil *library* sebuah komponen. Komponen yang dipanggil *library*-nya adalah *timer* dan LCD 16x2. Selain itu terdapat pendefinisian variabel-variabel yang akan digunakan untuk memudahkan dalam melakukan pemrograman.

### 3.4.2 Listing Program Setup

*Listing* program *setup* ini berfungsi sebagai inisialisasi *hardware* yang akan digunakan sesuai dengan kebutuhan, menentukan sebuah pin menjadi *input* atau *output*, seting pin dijadikan serial, atau akan digunakan LCD dll. *Listing* 3.2 menunjukkan *listing* program dari program *setup*.

```

// fungsi setup (hanya di jalankan 1x saat mikro pertama on)
void setup()
{
  // atur mode lcd 16x2
  lcd.begin(16,2);
  // atur pin i/o
  pinMode(ledpin, OUTPUT);
  pinMode(uvpin, OUTPUT);
  pinMode(tlpin, OUTPUT);
  pinMode(fan1pin, OUTPUT);
  pinMode(fan2pin, OUTPUT);
  pinMode(tombolpin, INPUT_PULLUP);
  // set off relay
  digitalWrite( uvpin, LOW );
  digitalWrite( tlpin, LOW );
  digitalWrite( fan1pin, LOW );
  digitalWrite( fan2pin, LOW );
  // aktifkan timer interrupt
  Timer1.initialize(1000000); // atur ke mode 1000000 microseconds ( 1s )
  // panggil program ISR
  Timer1.attachInterrupt( timerIsr );
}

```

**Listing 3.2** Listing Program Setup

Pada *Listing 3.2* dapat dijelaskan pada kode “*lcd.begin()*,” digunakan untuk menginisialisasi *library* LCD 16x2. Kode “*PinMode*” digunakan untuk melakukan konfigurasi secara spesifik fungsi dari sebuah pin apakah sebagai *input* atau *output*. Kode “*digitalWrite*” digunakan untuk memberikan nilai *high* atau *low* ke pin digital. Kode “*Timer.initialize*” dan “*Timer1.attachInterrupt*” digunakan untuk inisialisasi dan pemanggilan program *timer* ISR.

### 3.4.3 Listing Program Loop

*Listing* program *loop* merupakan fungsi utama yang akan berjalan terus menerus ketika program dijalankan. *Listing 3.3* menunjukkan *listing* program dari program *loop*

```

void loop()
{
  // jika tombol bernilai 0 timer mulai mencacah
  if(digitalRead(tombolpin)==0)
  {
    tanda_start=1;
  }
  else
  {
    tanda_start=0;
  }

  if(tanda_start==0)
  {
    counter=0;
    menit1=0;
    detik1=0;
    menit2=0;
    detik2=0;
    digitalWrite( uvpin, off );
    digitalWrite( tlpin, off );
    digitalWrite( fan1pin, off );
    digitalWrite( fan2pin, off );
  }
  // panggil baca sensor
  flow=baca_sensor();
}

```

**Listing 3.3** Listing Program Loop

Pada *Listing 3.3* dapat dijelaskan ketika “*tombolpin*” bernilai 1, maka sistem belum akan berjalan. Pada saat bernilai 0, “*tombolpin*” pada pin 12 *microcontroller* akan mendapatkan logika *low* sehingga akan menjalankan sistem.

Jika “*tanda\_start==0*” maka *timer* akan bernilai 0 dan semua pin “*digitalWrite*” akan mati. Kode “*baca\_sensor*” merupakan program untuk memanggil fungsi baca sensor, sehingga sensor akan mulai bekerja. Selanjutnya LCD akan menampilkan data yang diolah oleh *microcontroller*. *Listing 3.4* menunjukkan *listing* program dari program LCD.

```

// tampilkan data di lcd
if(counter==0)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("BIOLOGICAL");
  lcd.setCursor(0,1);
  lcd.print("SAFETY CABINET");
}

if(counter==1)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("AWAS SINAR UV");
  lcd.setCursor(6,1);
  lcd.print(menit1);
  lcd.print(":");
  lcd.print(detik1);
}

if(counter==2)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("STERILISASI");
  lcd.setCursor(6,1);

```

**Listing 3.4** Listing Program LCD

#### 3.4.4 Listing Program Timer ISR

*Timer* atau pewaktu merupakan program penting dalam sistem. Penggunaan *timer* dalam sistem bertujuan untuk membatasi atau memberi waktu kerja dari komponen yang berada pada sistem seperti lampu UV, lampu TL, dan fan. *Listing 3.5* menunjukkan *listing* program dari program *timer* ISR.

```

// fungsi timer ISR
// fungsi ini dipanggil di attach interrupt (program dalam mode interupsi)
void timerIsr()
{
    // timer aktif
    if(tanda_start==1)
    {
        // kedipan led penanda timer
        digitalWrite( ledpin, digitalRead( ledpin ) ^ 1 );
        // timer1 aktif
        if(menit1<timer1)
        {
            counter=1;
            detik1++;
            if(detik1>59)
            {
                menit1++;
                detik1=0;
            }
        }
        // jika timer 1 selesai
        if(menit1==timer1&&detik1==0)
        // timer2 aktif
        {
            if(menit2<timer2)
            {

```

**Listing 3.5** Listing Program Timer ISR

```

                counter=2;
                detik2++;
                if(detik2>59)
                {
                    menit2++;
                    detik2=0;
                }
                // uv on
                digitalWrite( uvpin, on );
            }
        }
        // jika timer2 selesai
        // uv off
        // t1, fan1, fan2, on
        if(menit2==timer2&&detik2==0)
        {
            counter=3;
            digitalWrite( uvpin, off );
            digitalWrite( t1pin, on );
            digitalWrite( fan1pin, on );
            digitalWrite( fan2pin, on );
            tanda_start=0; // reset tanda start
        }
    }
}

```

**Listing 3.5** Listing Program Timer (Lanjutan)

### 3.4.5 Listing Program Baca Sensor

Laju aliran udara yang melewati sensor akan diubah menjadi tegangan. Tegangan inilah yang digunakan dalam *listing* program baca sensor untuk dikonversi menjadi nilai laju aliran udara. *Listing 3.6* menunjukkan *listing* program dari program baca sensor.

```
// fungsi baca sensor
float baca_sensor()
{
  // nilai offset
  float minoffset=1.0,maxoffset=5.0;
  float flowsensor;
  // koversi dari nilai adc ke volt
  float vout=(float)analogRead(sensorpin)*((float)5/1023)-minoffset;
  vout=vout+minoffset;
  // ubah ke m/s
  if(vout<=(1.58-minoffset))flowsensor=vout*((float)1/(1.58-minoffset));
  if(vout>(1.58-minoffset)&&vout<=(2.88-minoffset))flowsensor=vout*((float)2/(2.88-minoffset));
  if(vout>(2.88-minoffset)&&vout<=(4.11-minoffset))flowsensor=vout*((float)3/(4.11-minoffset));
  if(vout>(4.11-minoffset))flowsensor=vout*((float)4/(5.0-minoffset));
  // pembatas minimal 0
  if(flowsensor<0)flowsensor=0;
  // nilai balik
  return flowsensor;
}
```

**Listing 3.6** Listing Program Baca Sensor

Pada *Listing 3.6* terdapat nilai *offset* yang merupakan tegangan yang timbul pada keluaran saat nilai *inputan*-nya nol. Tegangan ini digunakan untuk menentukan laju aliran udara terendah yang dapat dibaca oleh sensor. Kode “*minoffset=1.0*” berarti tegangan *offset* terendah adalah 1 volt, sedangkan kode “*maxoffset=5.0*” berarti tegangan *offset* tertinggi adalah 5 volt.

Setelah menentukan tegangan *offset* dari sensor, program selanjutnya adalah mengubah sinyal ADC (*analog to digital converter*) menjadi volt. Kode “*analogRead(sensorpin)*” merupakan perintah untuk membaca sinyal analog yang dihasilkan oleh sensor. Selanjutnya nilai sinyal tersebut akan dikalikan dengan hasil pembagian antara tegangan referensi sebesar 5 volt dan nilai resolusi ADC sebesar

1023 yang dikurangkan dengan nilai *minoffset*. Nilai volt yang dihasilkan akan di konversi lagi ke dalam satuan m/s, dimana satuan tersebut merupakan satuan kecepatan aliran udara.

### 3.5 Pengujian Alat

Pengujian merupakan langkah yang digunakan untuk mengetahui sejauh mana kesesuaian antara rancangan dengan kenyataan pada alat yang telah dibuat, apakah sudah sesuai dengan yang diharapkan atau tidak. Langkah-langkah pengujian alat ini dapat diuraikan sebagai berikut.

- a. Menyiapkan *laminar flow biological safety cabinet* dan peralatan yang dibutuhkan untuk pengujian.
- b. Menyiapkan lembar tabel pengukuran serta data pembanding berupa datasheet sensor yang digunakan pada *laminar flow biological safety cabinet*.
- c. Menghubungkan sumber tegangan dengan *laminar flow biological safety cabinet*.
- d. Menutup pintu *laminar flow biological safety cabinet*, kemudian nyalakan dengan menekan saklar *on*.
- e. Melakukan pengujian tahap pertama yaitu pengukuran *delay timer* selama 1 menit menggunakan *stopwatch*, kemudian catat hasil pengukuran pada lembar tabel pengukuran.
- f. Melakukan pengujian tahap kedua yaitu pengukuran *timer* sterilisasi selama 30 menit menggunakan *stopwatch*, kemudian catat hasil pengukuran pada lembar tabel pengukuran.

- g. Melakukan pengujian tahap ketiga yaitu pengukuran tegangan *output flow* sensor menggunakan multimeter. Pengambilan data pengukuran dilakukan tiap 2 menit selama 1 jam, sehingga akan menghasilkan 30 data. Catat hasil pengukuran pada lembar tabel pengukuran.
- h. Setelah semua tahap pengujian dilakukan dan lembar tabel pengukuran terisi, langkah berikutnya adalah analisis data hasil pengujian untuk memperoleh kesimpulan.