

## BAB III

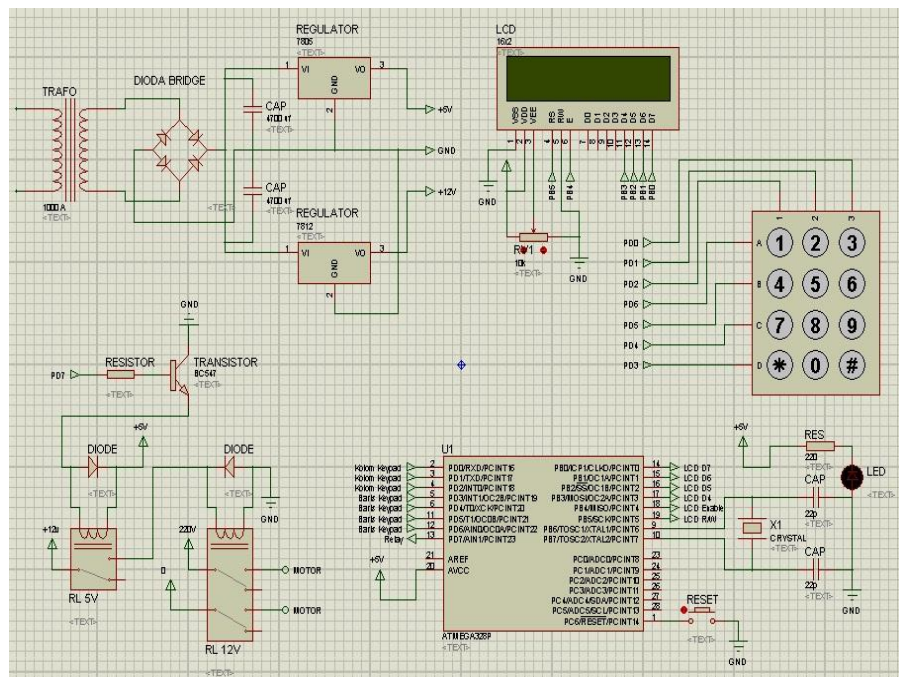
### METODE PENELITIAN

#### 3.1 Perancangann Alat

Perancangan alat adalah tahap dimana kita membuat atau merancang alat, mulai dari mengetahui alat dan bahan yang digunakan agar alat terancang seperti apa yang diharapkan.

Dalam perancangan alat perlu adanya langkah-langkah yang tepat agar hasil menjadi maksimal, langkah perancangan alat yaitu sebagai berikut :

##### 3.1.1 Perancangan *Hardware*

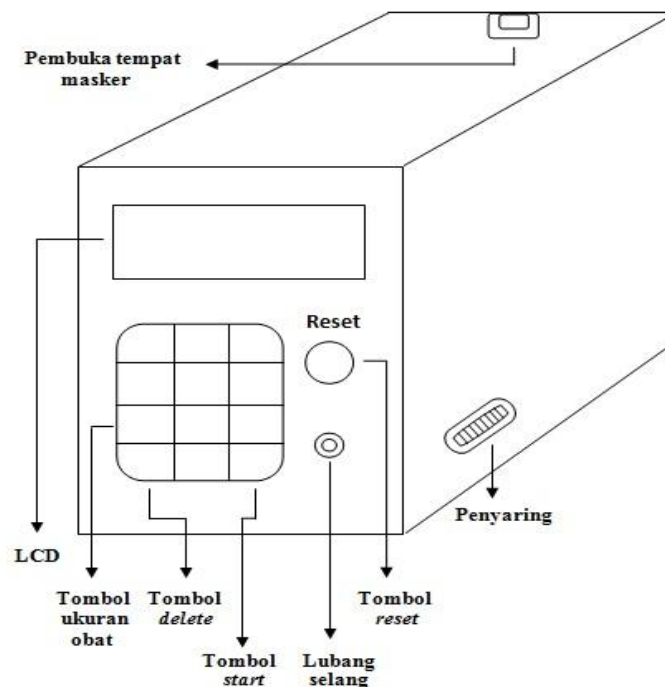


Gambar 3.1. Rangkaian keseluruhan

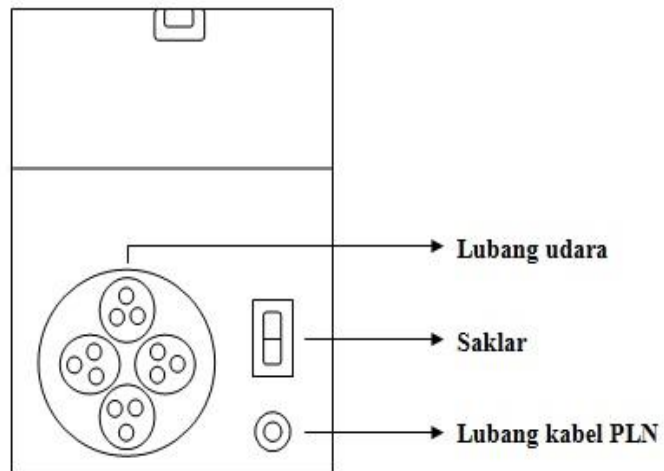
Perancangan *hardware* dilakukan dengan pembuatan skema rangkaian lengkap, dan juga perencanaan konstruksi dari alat tersebut.

### 1. Modifikasi *Nebulizer Compressor*

Kelemahan dari alat ini yaitu tidak adanya pemberitahuan otomatis jika obat telah habis sehingga harus dicek secara manual. Kondisi inilah yang melatarbelakangi untuk memodifikasi alat *nebulizer compressor* yang memiliki pengaturan waktu otomatis jika obat telah habis, lalu kompresor akan berhenti dengan otomatis. Pengaturan waktu otomatis dengan menggunakan rangkaian *driver motor* yaitu rangkaian yang didalamnya terdapat *relay* sebagai saklar.



Gambar 3.2. *Nebulizer Compressor* tampak depan



Gambar 3.3. *Nebulizer Compressor* tampak belakang

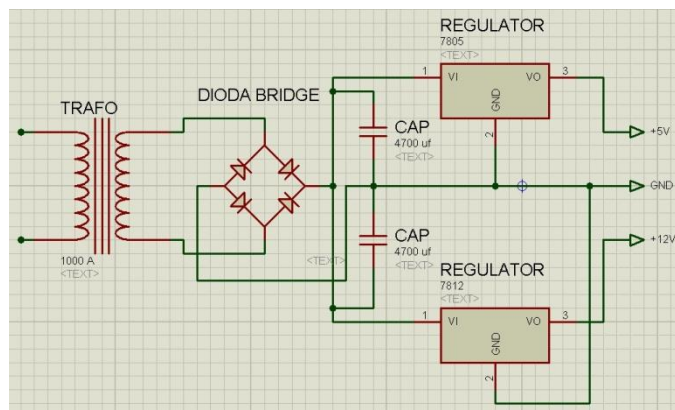


Gambar 3.4. *Nebulizer Compressor* modifikasi

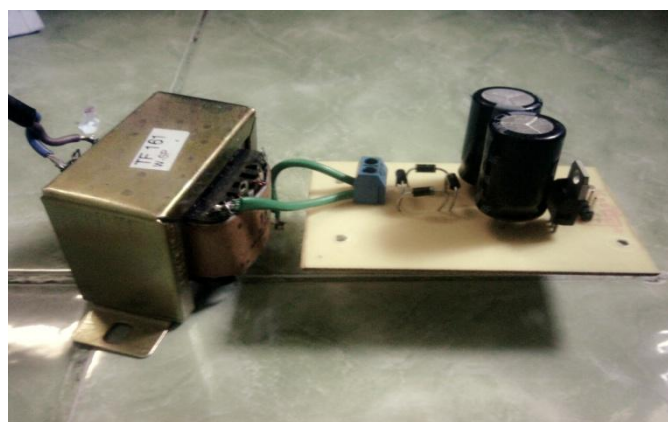
## 2. Rangkaian Catu Daya (*Power Supply*)

Rangkaian *Power Supply* berfungsi sebagai catu daya. Untuk mengetahui keluaran tegangan yang dihasilkan *power supply* dapat menggunakan multimeter. Multimeter dirubah ke mode ukuran *volt AC* agar dapat mengukur tegangan *AC* pada *power*

*supply*. Kemudian diubah lagi ke mode *volt DC* agar dapat mengukur tegangan *DC* pada *power supply*. Dari hasil pengujian *power supply* tegangan *DC* yaitu +5 volt, yang digunakan untuk memberi tegangan ke *relay 5 volt* dan *Display* sedangkan pengujian selanjutnya yaitu +12 volt dengan dua jumlah, yang digunakan untuk memberi tegangan pada rangkaian *minimum system* yang telah diberi *regulator 7805* dan untuk memberi tegangan *relay 12 volt*.



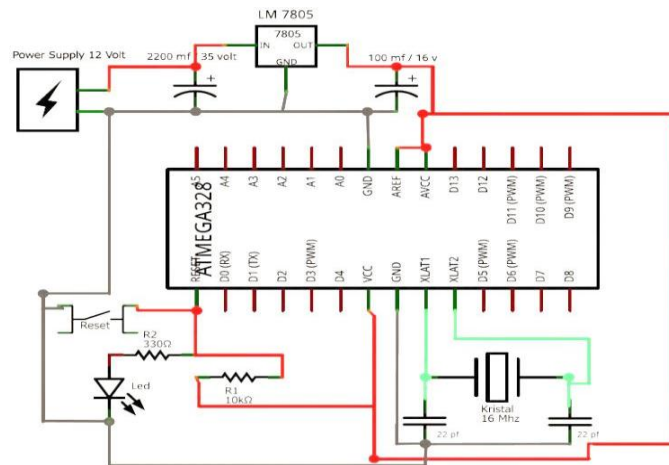
Gambar 3.5. Skematik Rangkaian *Power Supply*



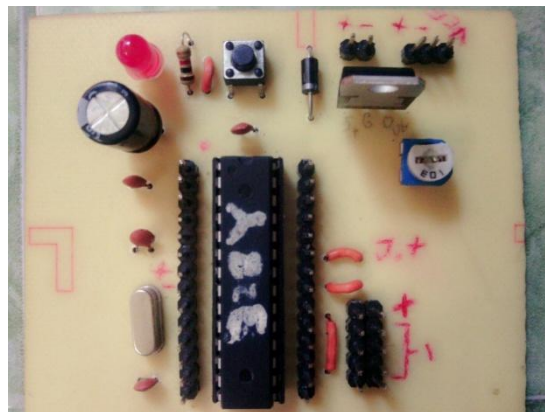
Gambar 3.6. Rangkaian *Power Supply*

### 3. Rangkaian *Minimum System ATmega 328*

Rangkaian *Minimum System* memiliki peran penting dalam seluruh sistem kerja alat.



Gambar 3.7. Skematik Rangkaian *Minimum System*



Gambar 3.8. Rangkaian *Minimum System*

Untuk mengetahui rangkaian Minimum Sistem dengan IC Mikrokontroler *ATmega 328* telah bekerja dengan benar maka dilakukan pengujian. Pengujian ini dengan menggunakan

program sederhana pada mikrokontroler *ATMega 328* yaitu

program *blink* pada *led*. Program sebagai berikut :

```

/*
  Blink
  Turns on an LED on for one second, then off for
  one second, repeatedly.
  This example code is in the public domain.
*/
// Pin 13 has an LED connected on most Arduino
boards.
// give it a name:
int led = 13;
// the setup routine runs once when you press
reset:

void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again
forever:

void loop() {
  digitalWrite(led, HIGH);    // turn the LED on
(HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by
making the voltage LOW
  delay(1000);                // wait for a second
}

```

Bila rangkaian *minimum system* telah berfungsi, *led* pada *project board* akan menyala secara blink dengan waktu 1 detik hidup lalu 1 detik mati secara berulang.

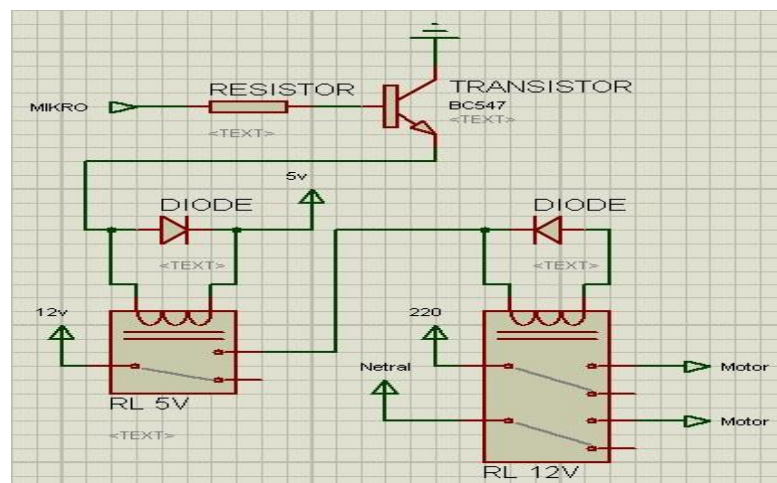
#### 4. Rangkaian *Driver Motor*

Rangkain *Relay* berfungsi sebagai saklar kontak pada motor. Dalam rangkaian ini terdapat 2 *relay* yang saling terhubung yaitu *relay 5 volt* dan *relay 12 volt*. Untuk menguji rangkaian driver motor, menggunakan *power supply* dan *minimum system*. Tahap pertama, rangkaian *relay* dihubungkan dengan *power*

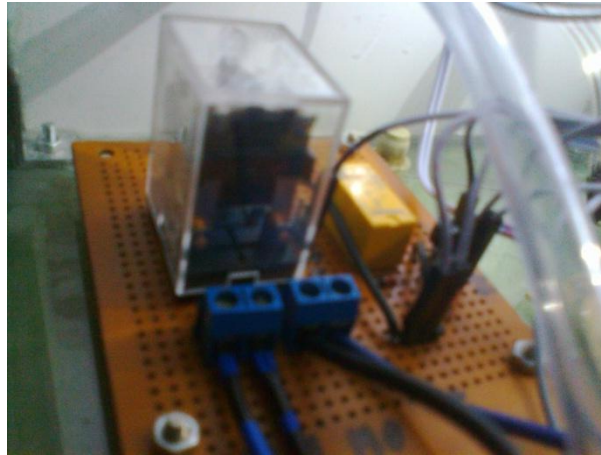
supply 12 volt dan 5 volt, lalu untuk *minimum system* dihubungkan ke 12 volt dan diisi program sebagai berikut :

```
int pin_relay = 7;
  if (timer_tenan != 0)
  {
    digitalWrite(pin_relay, HIGH);
  }
  else
  {
    digitalWrite(pin_relay, LOW);
  }
  delay(1000);
}
```

Jika kondisi *HIGH* maka *relay* akan mulai mengontak aktif motor untuk bergerak sedangkan kondisi *LOW* maka *relay* akan mengontak mati motor sehingga motor menjadi tidak aktif. *Pin* 7 di *arduino uno* sedangkan di *IC ATmega 328* menjadi *pin* ke 13.



Gambar 3.9. Skematik *Driver Motor*



Gambar 3.10. Rangkaian *Driver Motor*

#### 5. Rangkaian tampilan *Liquid Crystal Liquid (LCD)*

Rangkaian *LCD* berfungsi sebagai tampilan dengan berbagai macam karakter. Untuk menguji rangkaian tampilan pada *LCD* telah baik dan benar dilakukan pengujian. Tahap pertama, mengetahui fungsi masing-masing *pin* pada *LCD* yaitu seperti dibawah ini :

PIN	NAME	FUNCTION
1	VSS	Ground Voltage
2	VCC	5+ Voltage
3	VEE	Contrast Voltage
4	RS (Register Select)	0 = Register Select 1 = Data Register
5	R/W (Read/Write)	0 = Write Mode 1 = Read Mode
6	Enable	0 = Start to data LCD character 1 = Disable
7	DB0	Data Bus 0
8	DB1	Data Bus 1
9	DB2	Data Bus 2
10	DB3	Data Bus 3
11	DB4	Data Bus 4
12	DB5	Data Bus 5



13	<i>DB6</i>	<i>Data Bus 6</i>
14	<i>DB7</i>	<i>Data Bus 7</i>
15	<i>BPL</i>	<i>Black Panel Light</i>
16	<i>GND</i>	<i>Ground Voltage</i>

Setelah itu tahap kedua, yaitu mencoba meng-upload program dasar *LCD*, dengan cara merangkai *LCD* di *project board* lalu menggunakan *power supply* dan modul *minimum system* dengan disertai program *LCD "Hello Word"*, programnya yaitu sebagai berikut:

```

/*
  LiquidCrystal Library - Hello World

  Demonstrates the use a 16x2 LCD display.  The
  LiquidCrystal library works with all LCD displays that are
  compatible with the Hitachi HD44780 driver. There are many of them out
  there, and you can usually tell them by the 16-pin interface.

  This sketch prints "Hello World!" to the LCD
  and shows the time.

  The circuit:
  * LCD RS pin to digital pin 12
  * LCD Enable pin to digital pin 11
  * LCD D4 pin to digital pin 5
  * LCD D5 pin to digital pin 4
  * LCD D6 pin to digital pin 3
  * LCD D7 pin to digital pin 2
  * LCD R/W pin to ground
  * 10K resistor:
  * ends to +5V and ground
  * wiper to LCD VO pin (pin 3)

#include <LiquidCrystal.h>

// initialize the library with the numbers of the
// interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");

```

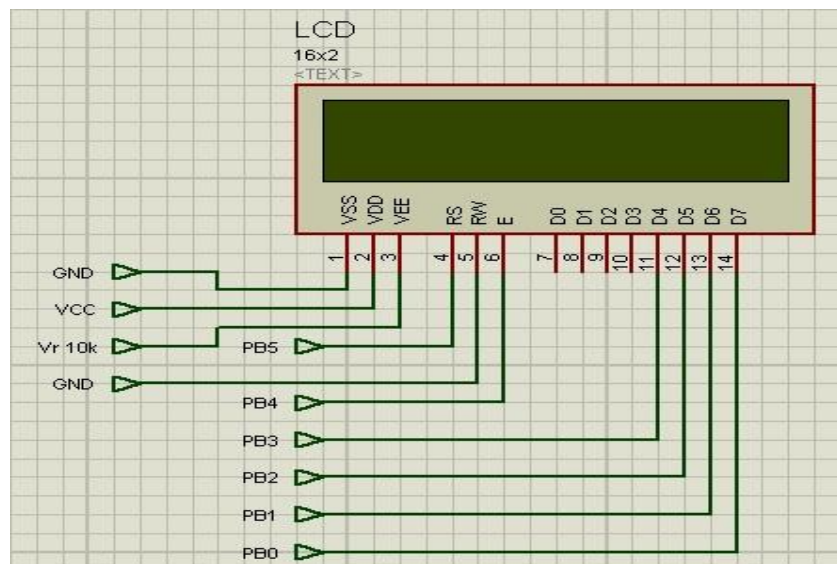
```

}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since
  counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}

```

Jika pada *LCD* menampilkan tulisan “*Hello Word*” maka rangkaian *LCD* telah berfungsi dengan benar.



Gambar 3.11. Skematik Rangkaian tampilan *LCD*

## 6. Peralatan dan bahan yang digunakan

Tabel 3.1. Alat dan Bahan

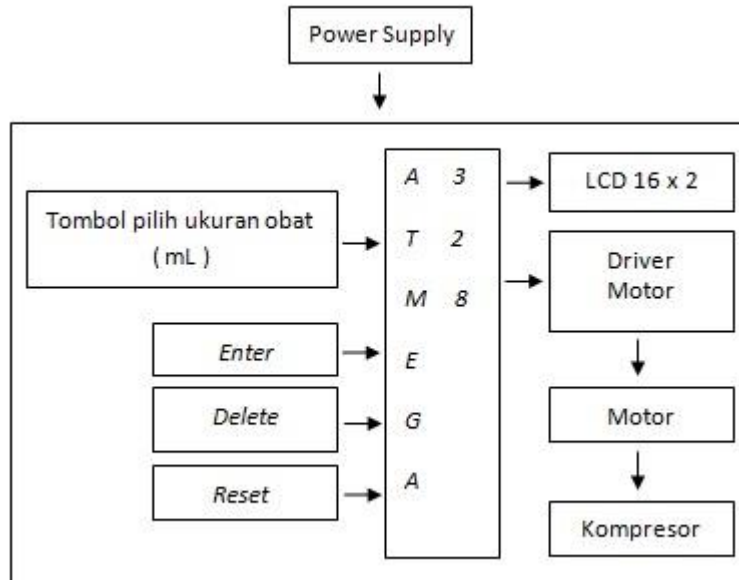
No	Komponen	Jumlah
1	<i>IC</i> Mikrokontroler <i>ATMega 328</i>	1
2	<i>Relay 5 Volt</i>	1
	<i>Relay 12 Volt</i>	1
4	Kompresor Udara	1
5	Motor AC 220 Volt	1

4	<i>Terminal Blox 2 pin</i>	3
5	<i>Resistor 1k</i>	1
	<i>Resistor 10k</i>	2
	<i>Resistor Variabel 10k</i>	1
6	<i>Capasitor 470 uF</i>	1
	<i>Capasitor 100 pF</i>	2
	<i>Capasitor 22 pF</i>	2
7	<i>Transistor BC547</i>	1
8	<i>Regulator IC 7805</i>	2
9	<i>Regulator IC 7812</i>	1
10	<i>Trafo 1000mA</i>	1
11	<i>Led</i>	1
12	<i>Diode IN4002</i>	6
14	<i>Push Button</i>	2
15	<i>Saklar On/Off</i>	1
16	<i>Black Housing</i>	Secukupnya

Sebagai sarana pendukung dalam pembuatan tugas akhir ini dapat kami sebutkan sebagai berikut :

1. Solder listrik
2. *Project board*
3. *Tool set*
4. Multimeter/Avometer digital
5. *Minidrill/Bor DC*

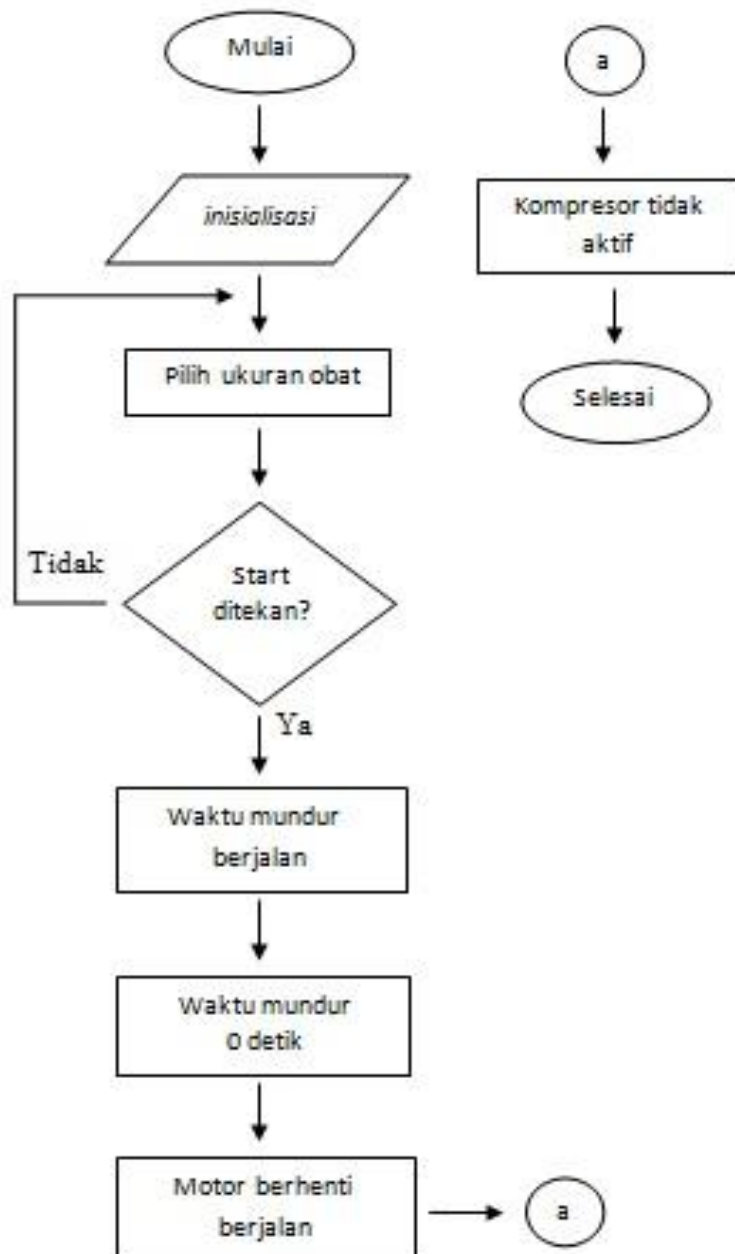
## 7. Pembuatan Blok Diagram



Gambar 3.12. Blok Diagram *Nebulizer Compressor*

Cara kerja blok diagram yaitu ketika pesawat *nebulizer* telah diatur ukuran obat dengan tombol *Keypad* kemudian *Enter* ditekan, dan (*counting down*) atau waktu mundur akan mulai aktif atau berjalan lalu *relay* akan mengontak motor sehingga motor aktif dan menggerakkan kompresor lalu bila waktu mundur menunjukkan 0 detik atau waktu telah habis, *relay* akan mengontak motor lagi menjadi tidak aktif sehingga kompresor berhenti dan alat tidak aktif.

## 8. Pembuatan Diagram Alir

Gambar 3.13. Diagram Alir *Nebulizer Compressor*

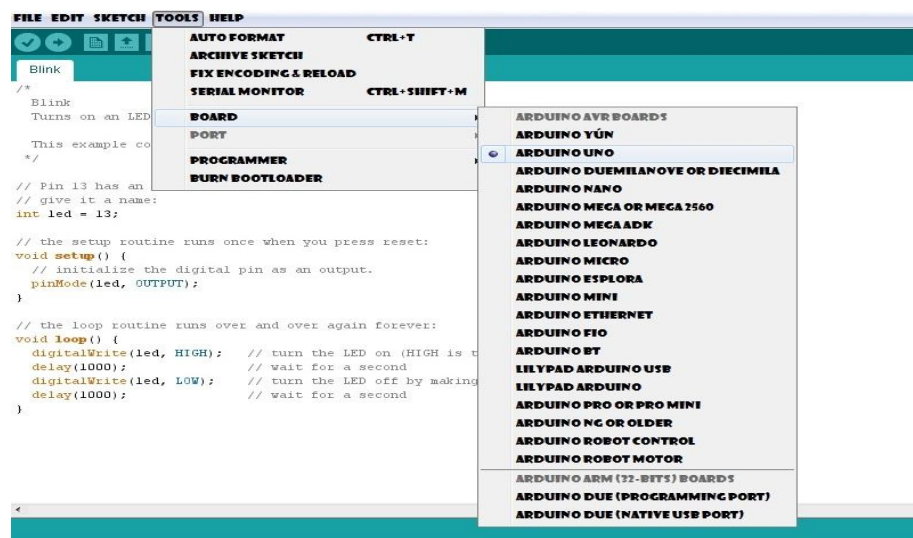
Diawali dengan memilih ukuran obat dengan tombol *Keypad* lalu menekan tombol *Enter*, dan (*counting down*) atau waktu

mundur akan mulai aktif atau berjalan lalu *relay* akan mengontak motor sehingga motor aktif dan menggerakkan kompresor lalu bila waktu mundur menunjukkan 0 detik, *relay* akan mengontak motor lagi menjadi tidak aktif sehingga kompresor berhenti dan alat tidak aktif. Bila ingin menjalankan pesawat seperti awal, tinggal menekan tombol *reset*.

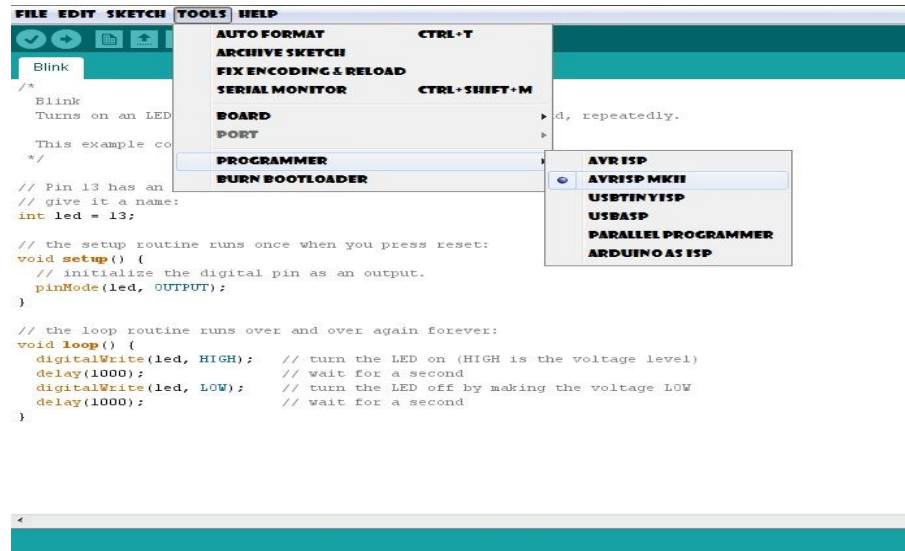
### 3.1.2 Perancangan *Software*

Pembuatan program dengan *software IDE (Integrated Development Environment) arduino uno*. Sebuah program harus terdiri dari 2 bagian, yaitu *inisialisasi* program dan program utama. *Inisialisasi* program harus disertakan agar program utama dapat berjalan. Berikut adalah urutan langkah *inisialisasi* program :

1. Menentukan jenis mikrokontroler yang akan digunakan dengan cara buka *software Arduino*, pilih *Tools* lalu *Board* kemudian pilih papan, karena menggunakan *ATMega 328* maka *board*-nya menggunakan *Arduino Uno*.

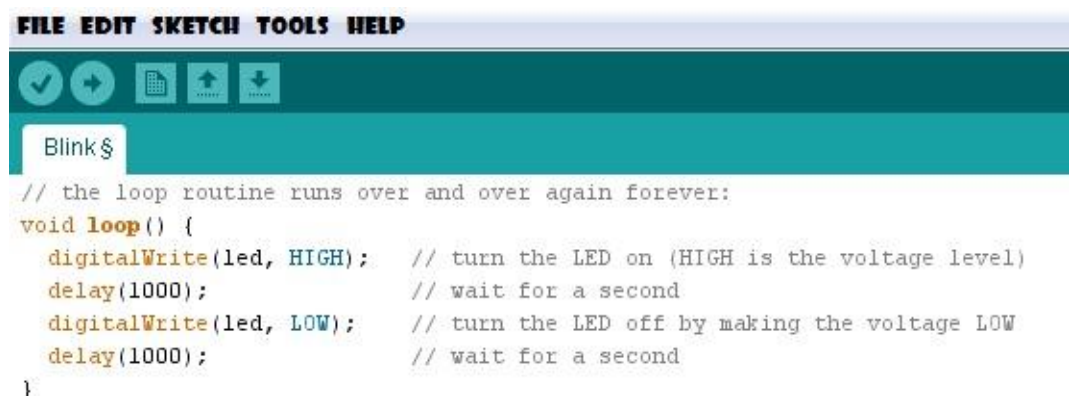


2. Menentukan *programmer* yang akan digunakan dengan cara pilih *tools* lalu *programmer* kemudian pilih *AVRISP mkII*.



3. Mengetahui program benar atau tidak dengan cara pilih *verify* lalu bila ingin meng-*upload* dengan cara pilih *upload* lalu tunggu beberapa saat sampai *loading* selesai, dan bila telah selesai akan bertuliskan *done uploading*

Lalu yang dimaksud dengan program utama yaitu program yang terus berulang-ulang (*loop*), gambarannya sebagai berikut :



Gambar 3.14. Contoh Program Utama

Keseluruhan program yang telah dibuat mulai dari program untuk *timer*, *driver motor* dan tampilan *LCD* disusun menjadi satu yaitu sebagai berikut :

Listining program

```
#include <LiquidCrystal.h>
#include <Keypad.h>

//Pendefinisian konstanta untuk keypad
const byte ROWS = 4;           //empat baris keypad
const byte COLS = 3;          //tiga kolom keypad
char keys[ROWS][COLS] = {     // Pemetaan keypad
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}}
};

// inisialisasi pin
LiquidCrystal lcd(13,12,11,10,9,8);
byte rowPins[ROWS] = {
  6, 5, 4, 3};                //Terhubung pinout baris
keypad
byte colPins[COLS] = {
  2, 1, 0};                    //Terhubung pinout kolom
keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins,
colPins, ROWS, COLS );

//deklarasi variabel
boolean angka_yang_ditekan = false;
boolean next = false;
boolean final = false;
String angka_kel1, angka_ke2;
int volume;
//=====
=====
void setup()
{
  // setup untuk awalan tampilan di LCD
  lcd.begin(16,2);

  lcd.setCursor(1,0);
  lcd.print(" YUDHISTIRA  ");
  lcd.setCursor(0,1);
  lcd.print(" 20113010013.  ");
  delay(3000);                //display text 3 dt
  lcd.clear();                //clear display
}
//=====
=====
```



```

void loop()
{
  char pencet = keypad.getKey();
  //set durasi
  lcd.setCursor(0,0);
  lcd.print(" ");
  lcd.setCursor(2,0);
  lcd.print("Set Volume : ");
  lcd.setCursor(8,1);
  lcd.print(" (ml)");
  lcd.setCursor(1,1);
  if (pencet != NO_KEY &&
(pencet=='1' || pencet=='2' || pencet=='3' || pencet=='4' || pe
ncet=='5' || pencet=='6' || pencet=='7' || pencet=='8' || pence
t=='9' || pencet=='0'))
  {
    if (angka_yang_ditekan != true)
    {
      lcd.setCursor(1,1);
      angka_kel = angka_kel + pencet;
      int dowone_karakter_bro = angka_kel.length();
      lcd.setCursor(4, 1);
      lcd.print(angka_kel);
    }
    else
    {
      angka_ke2 = angka_ke2 + pencet;
      int dowone_karakter_bro = angka_ke2.length();
      lcd.setCursor(15 - dowone_karakter_bro, 1);
      lcd.print(angka_ke2);
      final = true;
    }
  }

  //Set sistem !!
  if (pencet == '#') //Tombol enter
  {
    volume = angka_kel.toInt(); // mulai yg dimasukan
    adalah dalam ml
    int timer_tenan = ((volume * 2.5)+1) * 60; //
    persamaan dari  $y = 2,5x + 1$ 

    x (ml)    y (menit) (detik) //
                                                    //
    2         6         360 //
                                                    //
    4         11        660 //
                                                    //
    6         16        960 //
    for(int i = 0; i = timer_tenan ; i++)
    {
      timer_tenan--;
      int jam = timer_tenan / 3600;
      int menit = timer_tenan / 60;
      int detik = timer_tenan / 1;
      int detik_tenan = timer_tenan - menit * 60;

```

```
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Volume= ");
  if (volume>=10)
  {
    lcd.setCursor (8,0);
  }
  else
  {
    lcd.setCursor(8,0);
    lcd.print("0");
    lcd.setCursor(9,0);
  }
lcd.print(volume);
lcd.setCursor(11,0);
lcd.print("ml");
lcd.setCursor(0,1);
lcd.print("Waktu =");
  if (jam>=10)
  {
    lcd.setCursor (8,1);
  }
  else
  {
    lcd.setCursor(8,1);
    lcd.print("0");
    lcd.setCursor(9,1);
  }
lcd.print(jam);
lcd.setCursor(10,1);
lcd.print(":");
  if (menit>=10)
  {
    lcd.setCursor (11,1);
  }
  else
  {
    lcd.setCursor(11,1);
    lcd.print("0");
    lcd.setCursor(12,1);
  }
lcd.print(menit);
lcd.setCursor(13,1);
lcd.print(":");
  if (detik_tenan>=10)
  {
    lcd.setCursor (14,1);
  }
  else
  {
    lcd.setCursor(14,1);
    lcd.print("0");
    lcd.setCursor(15,1);
  }
lcd.print(detik_tenan);
```

```
int pin_relay = 7;
if (timer_tenan != 0)
{
    digitalWrite(pin_relay, HIGH);
}
else
{
    digitalWrite(pin_relay, LOW);
}
delay(1000);
}
lcd.clear();
lcd.setCursor(2,0);
lcd.print("=== HABIS ===");
timer_tenan = 0;
delay(5000);
}
//batal
else if (pencet == '*') //Tombol Delete
(backspace)
{
    lcd.clear();
    angka_yang_ditekan = false;
    final = false;
    angka_kel = "";
    angka_ke2 = "";
    volume = 0;
}
}
```