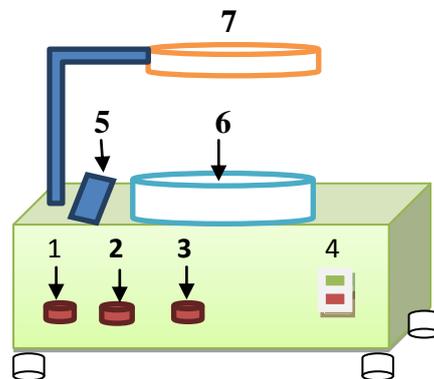


## BAB III METODOLOGI PENELITIAN

### 3.1 Diagram Mekanis Sistem

Sistem mekanis yang penulis buat menggunakan bahan plat logam yang sebelumnya telah dihaluskan dan melalui proses *quality control* sehingga diharapkan dalam penggunaannya *user* dapat lebih mudah menghitung, dan menganalisa hasil perhitungannya, *lup* yang berfungsi sebagai pembesaran memiliki spesifikasi 5 kali pembesaran dan juga *stand* yang berdiri dbuat seelastis mungkin sehingga *user* dapat menyesuaikan dalam proses pengoperasianya.



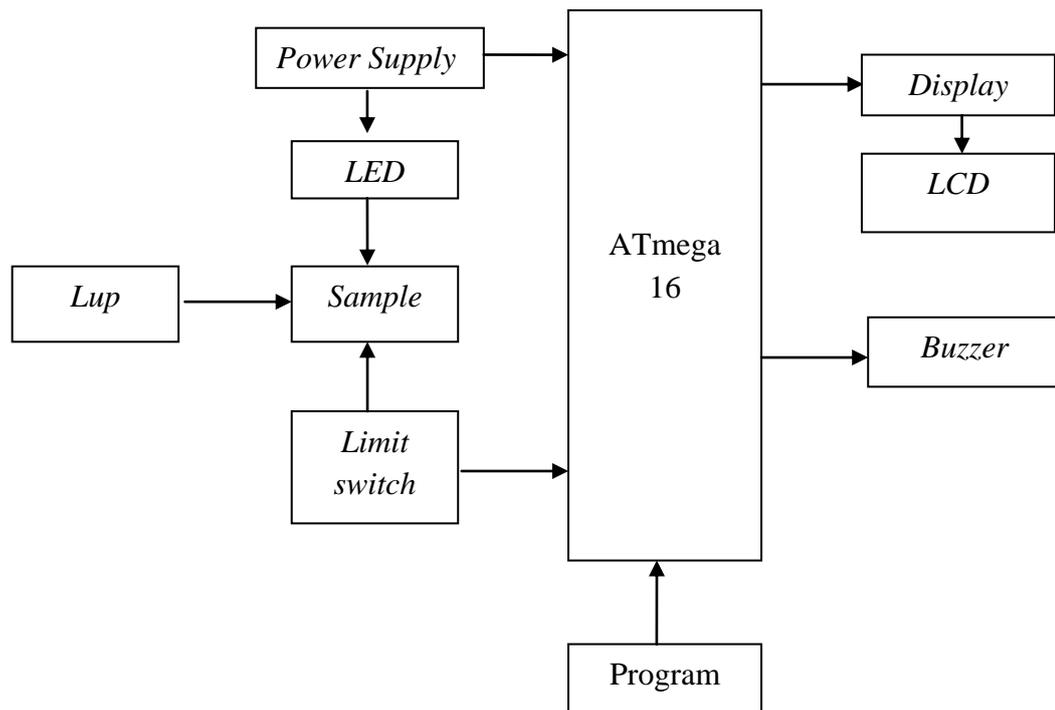
Gambar 3.1 Desain modul *colony counter*

Keterangan :

1. Tombol *up*
2. Tombol *down*
3. Tombol *start/reset*
4. Saklar *power*
5. *Display LCD*
6. Cawan petri
7. *Lup*

### 3.2 Blok Diagram

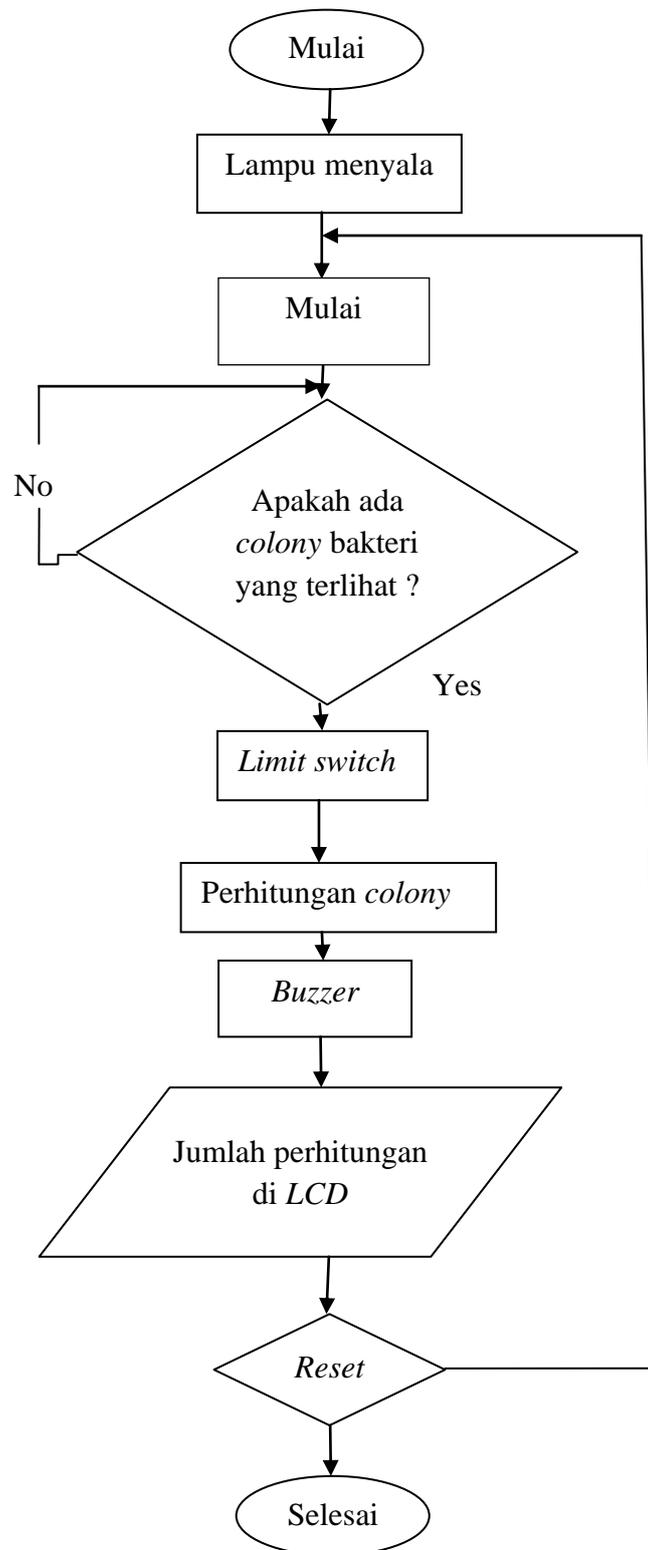
Pada saat saklar *ON* ditekan, tegangan dari jala-jala PLN akan masuk ke *power supply* untuk mengubah tegangan menjadi *DC*. *LED* akan menyala, *LED* berfungsi untuk menerangi objek. Pada saat objek ditekan *limit switch* mendapatkan sinyal dengan indikator *buzzer* bunyi, kemudian sinyal itu dikirim ke *microcontroller*, maka *microcontroller* memproses sinyal tersebut yang nantinya akan meng *counter* kemudian ditampilkan pada *LCD* sebagai *display*.



Gambar 3.2 Blok Diagram *Colony Counter*

### 3.3 Diagram Alir Program

Saat akan melakukan perhitungan *colony* siapkan spidol berukuran 0.2-0,8 sebagai penanda *colony*, tekan *colony* yang akan dihitung menggunakan spidol berukuran tertentu, sebagai indikator terhitungnya *colony buzzer* akan bunyi dan ditampilkan pada *LCD*. *Reset* untuk mengulangi perhitungan dari awal.



Gambar 3.3 Diagram Alir Program

### 3.4 Perakitan Rangkaian *Power Supply*

### 3.4.1 Alat

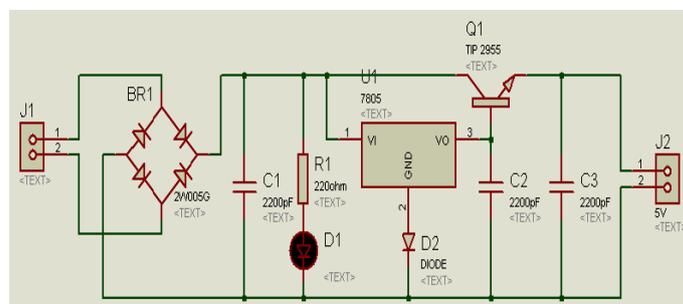
1. Papan *pcb*
2. Solder
3. Timah
4. Penyedot timah

### 3.4.2 Komponen

1. Transistor TIP 2955
2. Kapasitor 2200  $\mu\text{f}$
3. *IC regulator* 7805
4. *LED*
5. Dioda 1 A (4)
6. Travo 1 A
7. Resistor 220 *ohm*

### 3.4.3 Langkah Perakitan

1. Rangkai sistematis rangkaian *power supply* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.



Gambar 3.4 Rangkaian *power supply*

Arus *AC* akan diubah menjadi arus *DC* untuk mengubah arus *DC* tersebut menjadi gelombang penuh maka dipasang dioda bridge atau dioda *rectifier*. Gelombang yang sudah menjadi gelombang penuh akan diberi filter berupa kapasitor yang dipasang secara paralel dengan nilai 2200  $\mu\text{f}$  nilai ini tidak menjadi acuan berapapun nilai kapasitor yang digunakan, namun penulis lebih memilih nilai tersebut. Setelah mendapat filter dari kapasitor gelombang menjadi berbentuk *ripple*. Sehingga gelombang belum. Dikatakan setabil atau belum berbentuk gelombang *DC* sempurna. Maka dipasang *LED* indikator sebagai tanda gelombang *DC* terbentuk, agar arus *LED* tidak berlebihan maka dipasang resistor dengan nilai 220 *ohm*, berikut perhitungannya

Berikut ini adalah cara menghitung nilai resistor yang diperlukan untuk rangkaian *LED* agar tidak mengalami kerusakan karena kelebihan arus dan tegangan.

Rumus yang dipakai adalah sebagai berikut :

$$R = (V_S - V_L) / I$$

Dimana :

$R$  = Nilai resistor yang diperlukan (dalam *ohm*)

$V_S$  = Tegangan *input* (dalam *volt*)

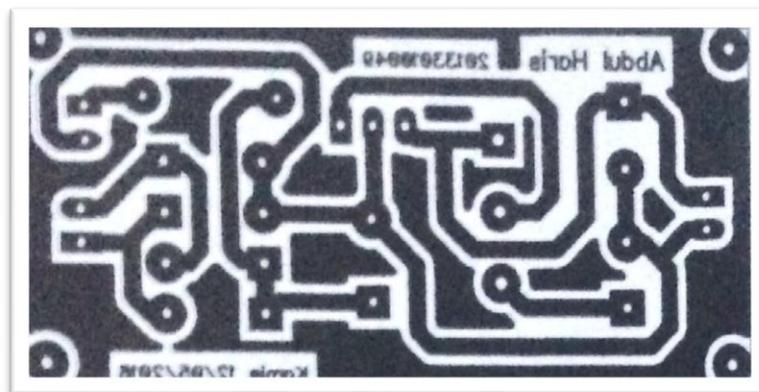
$V_L$  = Tegangan *LED* (dalam *volt*)

$I$  = Arus maju *LED* (dalam *Ampere*)

$$\begin{aligned}
 R &= \frac{6-1,7}{20} \\
 &= \frac{4,3}{0,02} \\
 &= 215 \text{ ohm}
 \end{aligned}$$

Karena nilai 215 *ohm* tidak ada dalam *datasheet* maka dipasang nilai yang mendekati 215 *ohm* yaitu 220 *ohm*. 6 volt DC diturunkan menjadi 5 volt DC oleh IC regulator 7805 untuk pengaman regulator maka dipasang dioda pada pin 3 kemudian 5 volt DC dari regulator masuk ke transistor TIP yang berfungsi sebagai penguat arus, kemudian arus 5 volt DC stabil diperkecil menggunakan 2 kapasitor yang dipasangkan secara paralel dengan nilai tertentu sesuai kebutuhan dalam rangkaian ini penulis memasangkan dengan nilai 2200  $\mu\text{f}$ .

2. Setelah sistematis rangkaian jadi, tahap selanjutnya membuat *lay out* nya dan disablon ke papan *pcb*. Untuk gambar *lay out power supply* pada papan *pcb* dapat dilihat pada gambar 3.4 di bawah ini :

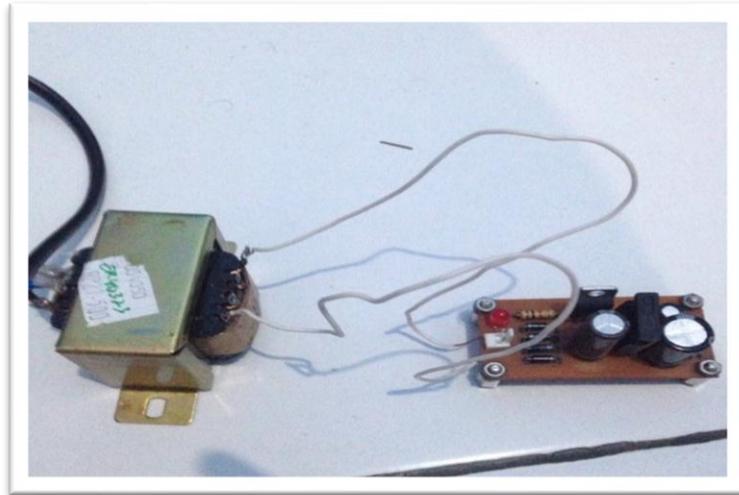


Gambar 3.5 *Lay out power supply*

3. Rakit komponen yang dibutuhkan dengan menggunakan solder.

### 3.4.4 Gambar *Power supply*

Untuk gambar *power supply* dapat dilihat pada gambar 3.6 bawah ini:



Gambar 3.6 *Power Supply*

Rangkaian *power supply* pada modul ini berfungsi sebagai *supply* tegangan ke semua rangkaian yang menggunakan tegangan *DC*. Prinsip kerja *power supply* adalah mengubah tegangan *AC* menjadi tegangan *DC* dengan menggunakan *transformator* sebagai penurun tegangan dan dioda sebagai komponen yang berfungsi sebagai penyearah tegangan. Pada modul ini *power supply* akan mengubah tegangan *AC* menjadi *DC* sebesar 5 *VDC* dengan menggunakan *IC regulator 7805*. Tegangan 5 *VDC* digunakan untuk rangkaian minimum sistem.

### **3.5 Perakitan Rangkaian Minimum Sistem**

#### **3.5.1 Alat**

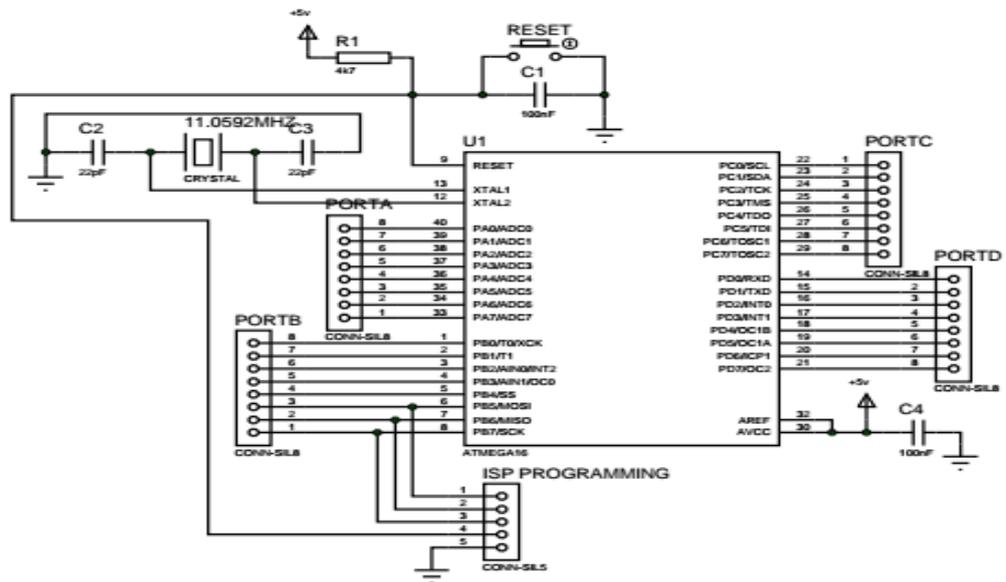
1. Papan *pcb*
2. Solder
3. Timah
4. Penyedot timah

#### **3.5.2 Komponen**

1. ATmega 16
2. Kapasitor 10  $\mu$ f 25 v
3. Kapasitor *non* polar
4. *Crystal* 16 Mhz

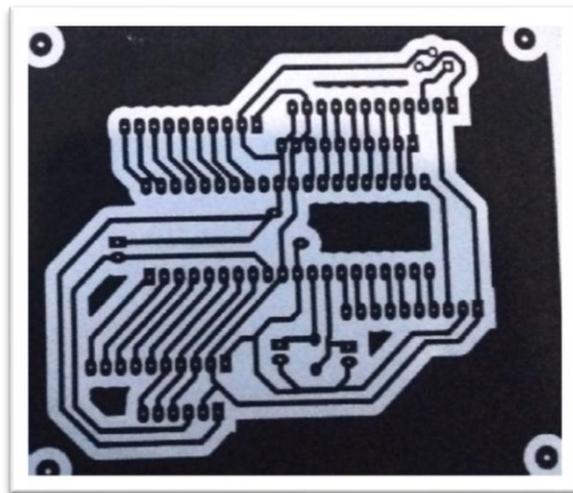
#### **3.5.3 Langkah Perakitan**

1. Rangkai minimum sistem dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.
2. Untuk gambar sistematik rangkaian minimum sistem pada aplikasi dapat dilihat pada gambar 3.7 di bawah ini:



Gambar 3.7 Rangkaian Minimum Sistem

3. Setelah sistematis rangkaian jadi, tahap selanjutnya membuat *lay out* nya dan disablon ke papan *pcb*. Untuk gambar *lay out* minimum sistem pada papan *pcb* dapat dilihat pada gambar 3.8 di bawah ini:

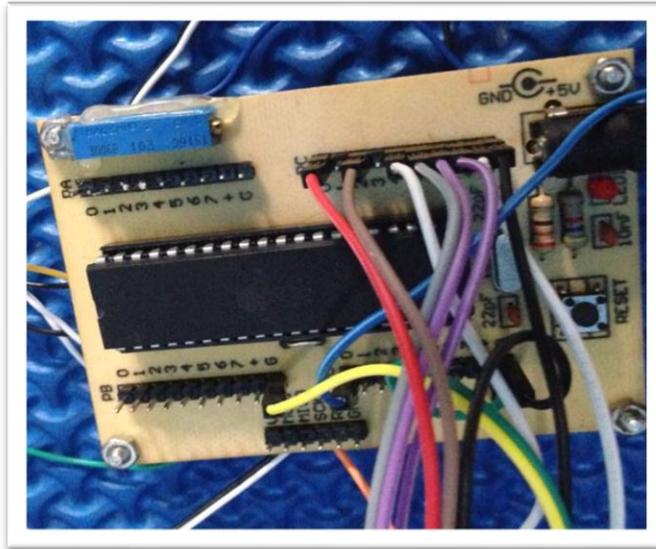


Gambar 3.8 *Lay out* minimum sistem

4. Rakit komponen yang dibutuhkan dengan menggunakan solder.

### 3.5.4 Gambar Minimum Sistem

Untuk gambar minimum sistem dapat dilihat pada gambar 3.9 di bawah ini:



Gambar 3.9 Minimum Sistem

Rangkaian minimum sistem pada modul ini berfungsi sebagai kontrol kerja modul secara keseluruhan. Cara kerja rangkaian minimum sistem ini dengan memanfaatkan kapasitas penyimpanan yang dimiliki oleh IC ATmega 16. Pada IC ATmega 16 ini diberi program yang akan mengontrol sistem kerja modul secara keseluruhan. Adapun program yang digunakan pada modul ini adalah *counter* sebagai perhitungan *colony*.

### **3.6 Perakitan rangkaian LCD**

#### **2.6.1 Alat**

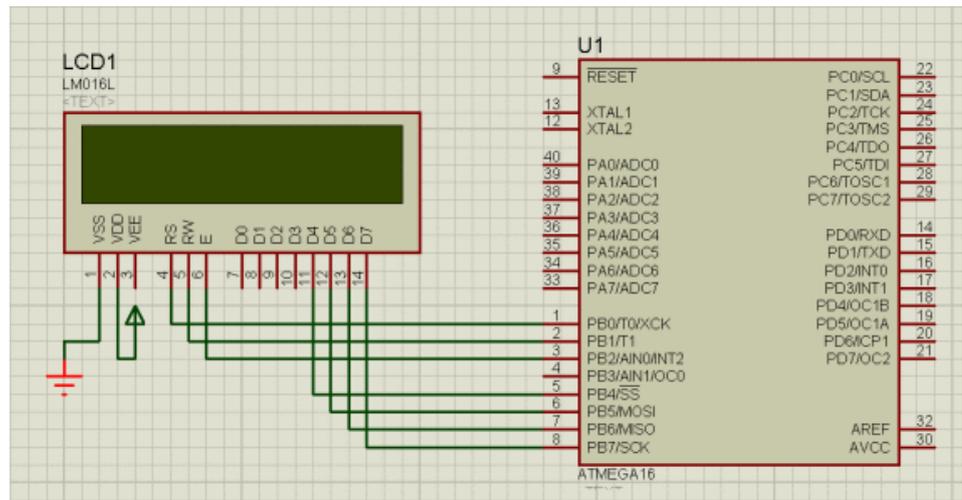
1. kabel pelangi
2. Solder
3. Timah
4. Penyedot timah
5. Soket *male female*

#### **2.6.2 Komponen**

1. LCD 2 x 16
2. Kabel pelangi

#### **2.6.3 Langkah perakitan**

1. Rangkai sistematis rangkaian LCD dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.
2. Untuk gambar sistematis rangkaian LCD pada aplikasi dapat dilihat pada gambar 3.10 di bawah ini



Gambar 3.10 Rangkaian *LCD* pada *proteus*

3. Rakit komponen yang dibutuhkan dengan menggunakan solder.
4. Rangkaian *LCD* pada modul alat *colony counter* disini berfungsi sebagai display, prinsip kerja dari rangkaian *LCD* adalah *ground* dan *vcc* yang terdapat pada *LCD* langsung mendapatkan tegangan dari sumber dan pada kaki *RS* diset dengan logika 1 sehingga akan memproses dan melakukan pengiriman pada program *ASCII*, *RW* diset dengan logika “0” dan D4, D5, D6, D7, mengirimkan data secara paralel 4 bit.

### 3.7 Perakitan Rangkaian *LED*

#### 3.7.1 Alat

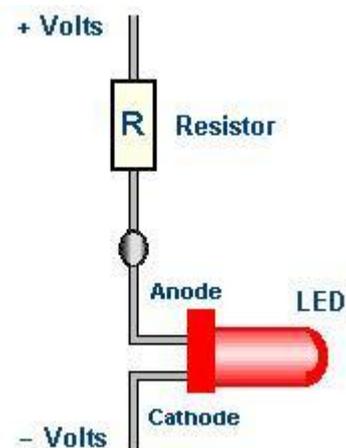
1. kabel pelangi
2. Solder
3. Timah
4. Penyedot timah

### 3.7.2 Komponen

1. *LED*
2. Resistor 220 *ohm*

### 3.7.3 Langkah Perakitan

1. Rangkai sistematis rangkaian *LED* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.
2. Untuk gambar rangkaian *LED* dapat dilihat pada gambar 3.11 di bawah ini :



Gambar 3.11 Rangkaian *LED*

3. Berikut ini adalah cara menghitung nilai resistor yang diperlukan untuk rangkaian *LED* agar tidak mengalami kerusakan karena kelebihan arus dan tegangan.

Rumus yang dipakai adalah sebagai berikut :

$$R = (V_S - V_L) / I$$

Dimana :

$R$  = Nilai resistor yang diperlukan (dalam *ohm*)

$V_S$  = Tegangan *input* (dalam *volt*)

$V_L$  = Tegangan *LED* (dalam *volt*)

$I$  = Arus maju *LED* (dalam *Ampere*)

$$\begin{aligned} R &= \frac{5-0,7}{0,02} \\ &= \frac{4,3}{0,02} \\ &= 215 \text{ ohm} \end{aligned}$$

Karena nilai 215 *ohm* tidak ada dalam *datasheet* maka dipasang nilai yang mendekati 215 *ohm* yaitu 220 *ohm*

4. Rakit komponen yang dibutuhkan dengan menggunakan solder.
5. Rangkaian *LED* disini berfungsi untuk menerangi objek prinsip kerja dari rangkaian *LED* ketika kaki positif mendapatkan tegangan 5 *volt* dari *power supply vcc* dan kaki negatif mendapatkan *ground LED* akan menyala.

### 3.8 Rangkaian *buzzer*

#### 3.8.1 Alat

1. Kabel pelangi
2. Solder
3. Timah
4. Penyedot timah

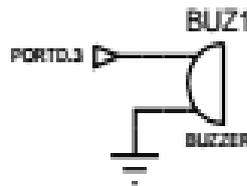
#### 3.8.2 Komponen

1. *Buzzer*
2. Resistor 1 k (1)

3. Resistor 4,7 k (1)
4. Transistor

### 3.8.3 Langkah Perakitan

1. Rangkai sistematis rangkaian *buzzer* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.
2. Untuk gambar rangkaian *buzzer* dapat dilihat pada gambar 3.12 di bawah ini



Gambar 3.12 Rangkaian *buzzer*

3. Rakit komponen yang dibutuhkan dengan menggunakan solder.
4. Rangkaian *buzzer* disini berfungsi sebagai indikator bahwa *colony* bakteri telah terhitung, prinsip kerja dari rangkaian *buzzer* ketika dari *portD* mendapatkan (+) kaki buzzer dan pada kaki (-) buzzer langsung mendapatkan *ground*.

## 3.9 Rangkaian *Limit Switch*

### 3.9.1 Alat

1. Kabel pelangi
2. Solder
3. Timah

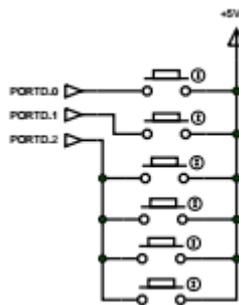
4. Penyedot timah

### 3.9.2 Komponen

1. *Limit switch* (4)
2. Saklar *push ON* (2)

### 3.9.3 Langkah perakitan

1. Rangkai sistematis rangkaian *limit switch* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.
2. Untuk gambar rangkaian *limit switch* dapat dilihat pada gambar 3.13 di bawah ini



Gambar 3.13 Rangkaian *limit switch*

3. Rangkaian *limit switch* disini berfungsi sebagai sensor mekanik yang memberikan sinyal pada *microcontroller* yang kemudian diproses dan dapat ditampilkan pada display *LCD*.

### 3.10 Pembuatan program *counter*

Untuk pembuatan program pada modul ini menggunakan aplikasi AVR dengan bahasa C. Program yang digunakan ialah program *counter* sebagai pengendali perhitungan *colony*.

Program *counter*

```
#include <mega16.h> //preprosesor
#include <delay.h>
#include <stdlib.h>
#include <alcd.h>
int hitung=0; //menyimpan dalam int
unsigned char temp[6];
void main(void)
{
PORTA=0x00;
DDRA=0x00;
PORTB=0x0F;
DDRB=0x00;
PORTC=0x00;
DDRC=0x00;
PORTD=0x00;
DDRD=0xFF;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
MCUCR=0x00;
```

*Listing 3.1* program *counter*



```

        if(hitung>=9999)
        {
            hitung=9999;
        }
        if(hitung<0)
        {hitung=0;}
        if(PINB.2==0)
        {hitung=0;lcd_clear();}
        itoa(hitung,temp);// menampilkan
interger to ascii
        lcd_gotoxy(8,1);
        lcd_puts(temp);//mengambil nilai
string temp
    }}

```

*Listing 3.3* program menampilkan *LCD*

Penjelasan program :

Dalam pembuatan program ini, penulis hanya memanfaatkan fasilitas *input-output* sebagai *counter up* dan *counter down*. Sedangkan tampilan penulis menggunakan fasilitas *LCD* sebagai keluaran.

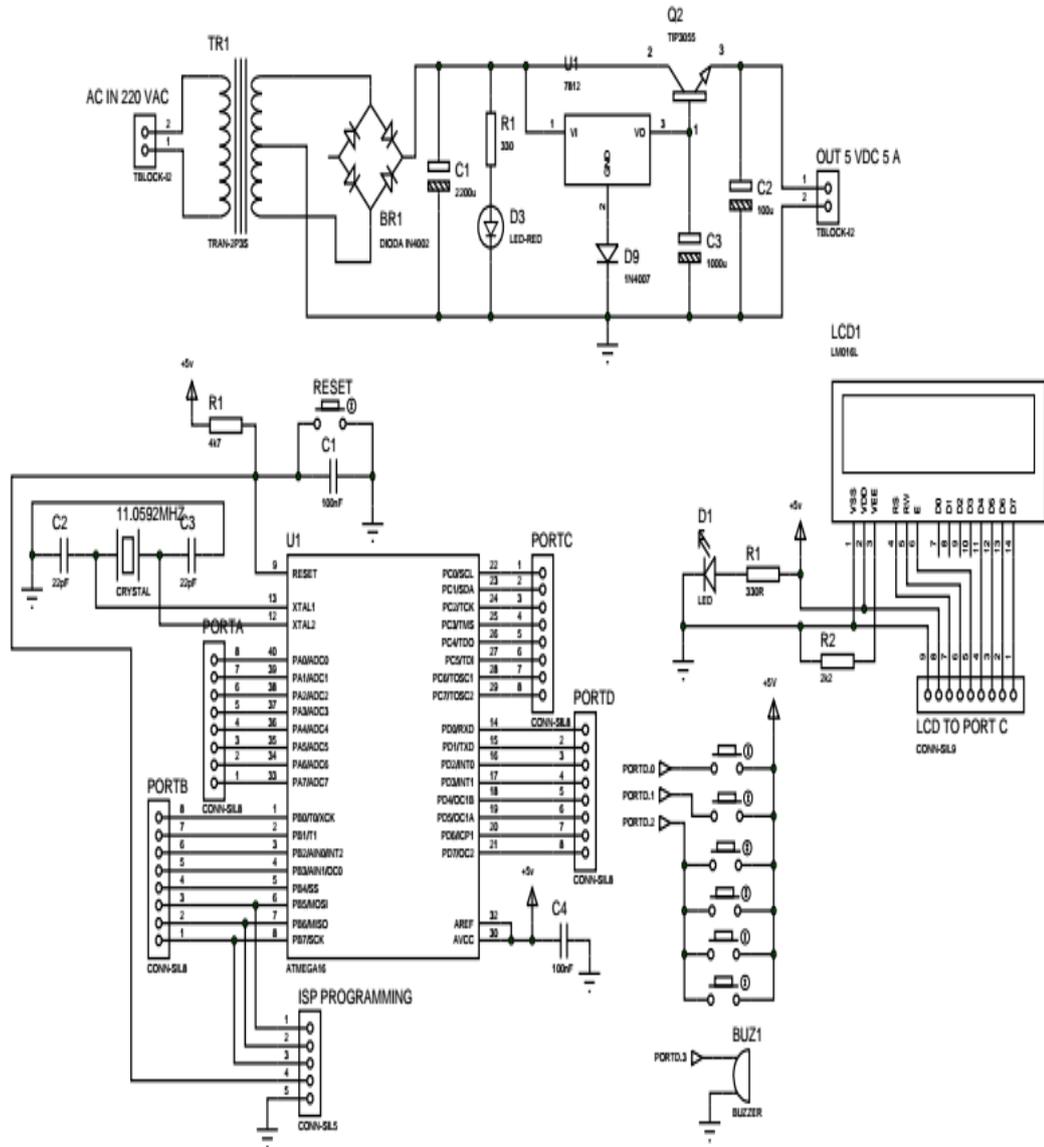
Cara kerja program ini sederhana. Setelah inisialisasi subrutin program akan menampilkan tampilan *LCD* berupa “*Colony Counter*” dan “Abdul Haris”. Setelah itu masuk ke inti program, dan menampilkan “*Colony Counter*” jika  $PINB.2==1$  maka akan menampilkan “*Colony Counter*” dan di *looping* sampai ada *input* di  $PINB.1$  atau di  $PINB.0$ . Jika terjadi *input*  $PINB.0==0$  atau terjadi *counter* maka nilai akan bertambah sesuai penekanan diikuti *buzzer* berbunyi. Jika tidak maka masih proses *looping* di  $PINB.2==1$  dan *buzzer OFF*. Jika  $PINB.1==0$  atau terjadi *input* maka akan terjadi *counter* turun (pengurangan) dan *buzzer on*. Jika perhitungan sampai nilai 9999 maka akan direset kembali menjadi 9999. Namun jika nilai  $<0$  maka akan direset menjadi 0. Jika *input*  $PINB.2==0$

maka layar akan direset ulang sehingga nilai tertera akan dihapus sesuai nilai awal atau kosong dan ditampilkan pada *LCD*.

### **3.11 Rangkaian keseluruhan**

Rangkaian ini tersusun dari beberapa blok-blok *PCB* yang sudah terpasang komponen-komponen sesuai fungsi dari blok tersebut dan di jadikan satu secara elektrik agar menjadi sebuah sistem yang dapat di gunakan sesuai maksud perancang modul. Ada beberapa blok dan rangkaian komponen yang terpasang dalam satu sistem ini antara lain adalah :

1. *Block Power suplay.*
2. *Block Rangkaian limit switch.*
3. *Block Minimum sistem.*
4. Rangkaian *LCD*.
5. Rangkaian *LED*.
6. Rangkaian *Buzzer*.



Gambar 3.14 Rangkaian Keseluruhan