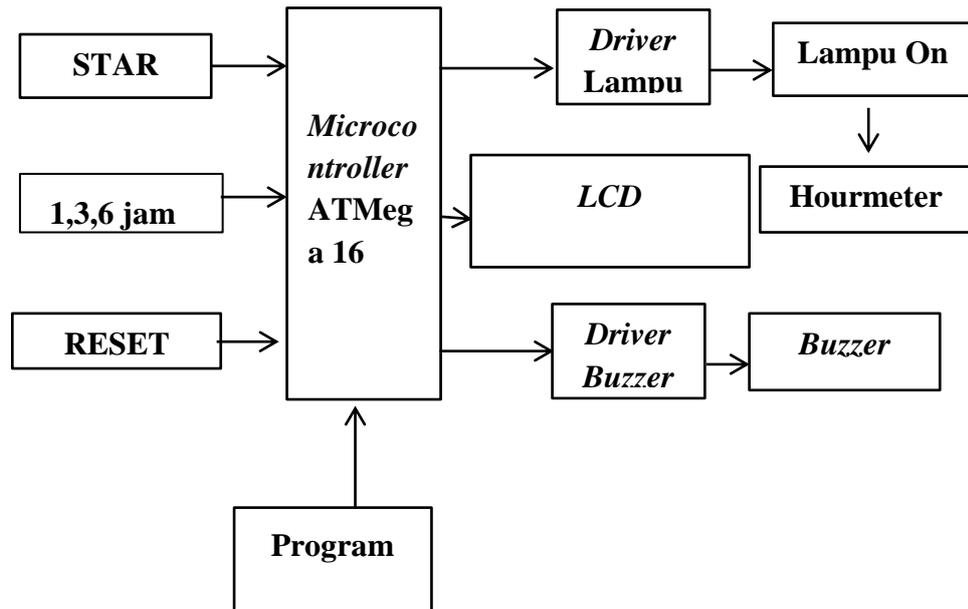


### BAB III METODE PENELITIAN

#### 3.1. Diagram Blok *Sterilisator UV*

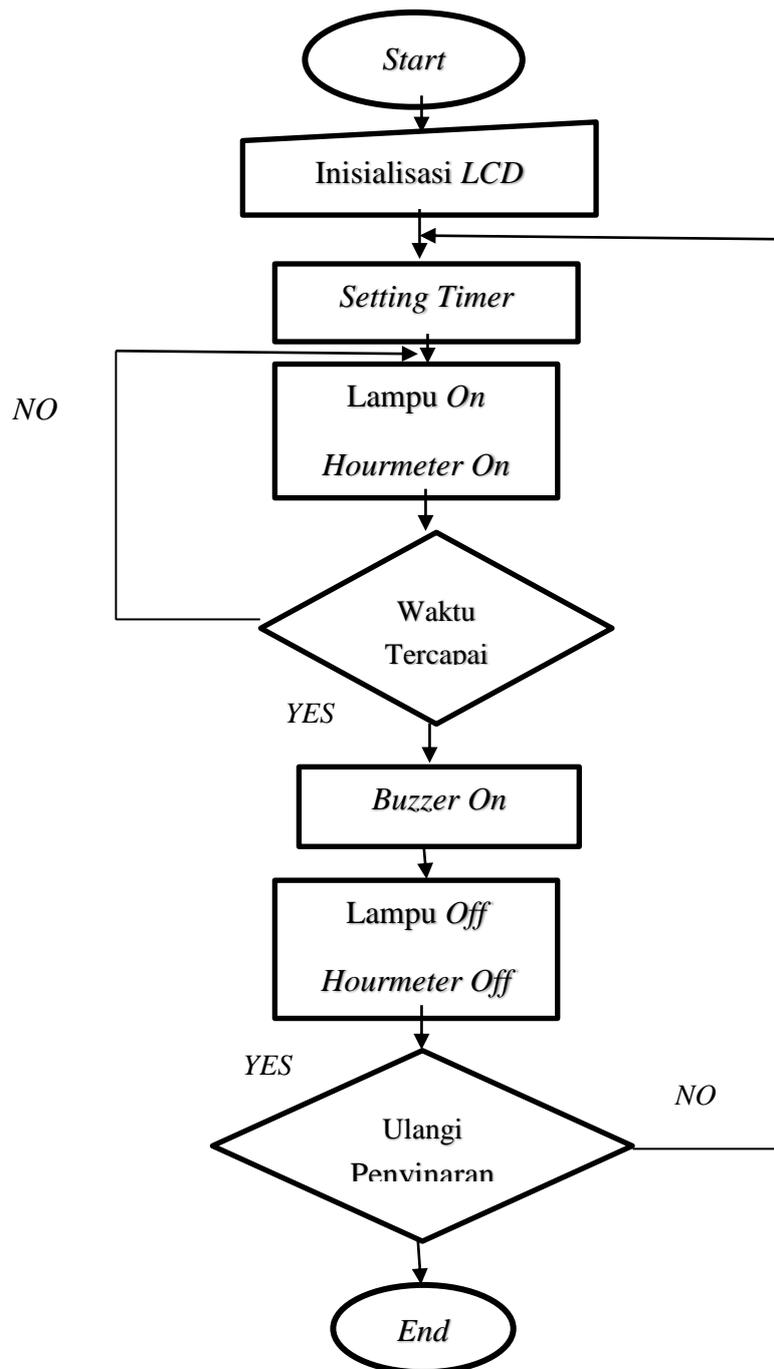


Gambar 3.1 Diagram Blok *sterilisator UV*

Tombol *ON/OFF* digunakan untuk menghidupkan atau mematikan. *Timer* ditentukan menggunakan tombol *timer* dengan pilihan waktu *sterilisator* selama 1 jam, 3 jam dan 6 jam. *Timer* persiapan waktu sudah ditentukan oleh *microcontroller* selama 5 menit. *Timer* ditampilkan dengan *LCD* 16x2. Kemudian tekan tombol *START* untuk memulai *sterilisator*. *Microcontroller* akan mengirimkan data untuk mengaktifkan *driver* agar menyalakan lampu *UV* dan *hourmeter*. *Timer* akan menghitung *counter*

*down* dan *hourmeter* menghitung *counter up* saat waktu habis lampu *UV* mati, *hourmeter* juga berhenti berjalan dan *buzzer* akan bunyi.

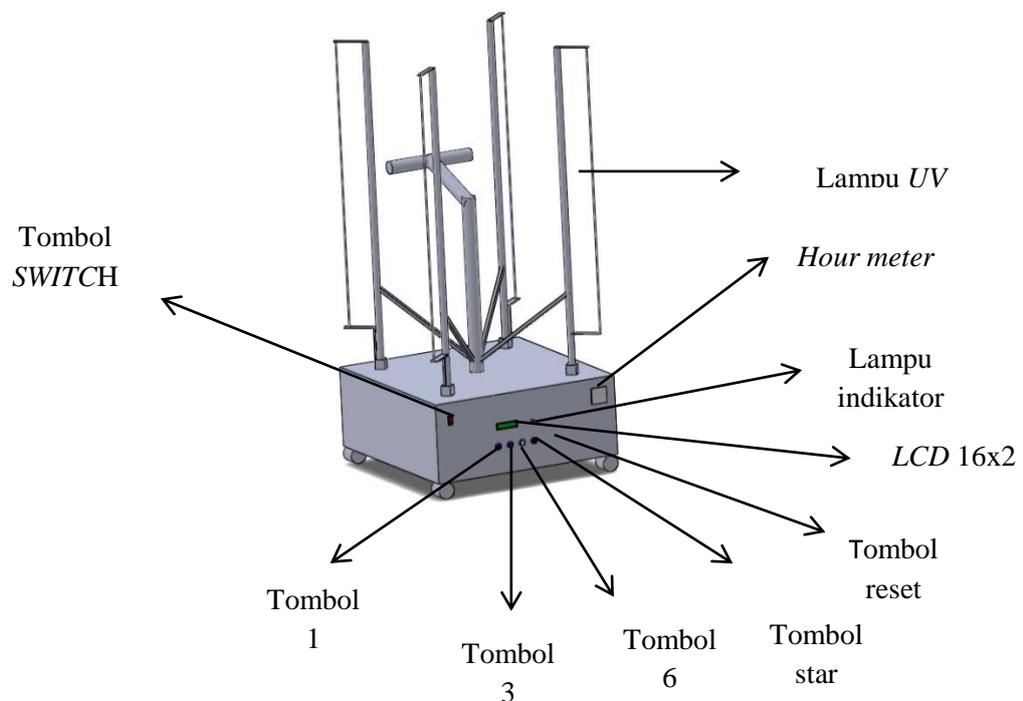
### 3.2. Blok Diagram Alir *Sterilisator UV*



Gambar 3.2. Diagram Alir *sterilisator UV*

Pertama tekan *STAR* proses inialisasi dilakukan pembacaan. Setelah itu lakukan pemilihan *setting timer* untuk lama *sterilisator* dengan menekan tombol 1 jam, 3 jam dan 6 jam. Persiapan waktu berjalan selama 5 menit. Lalu lampu akan menyala bersamaan dengan *timer* akan bekerja *mengcounter down* dan *hourmeter* akan bekerja *mengcounter up* jika waktu masih berjalan belum tercapai kembali lagi lampu tetap *on hourmeter on NO*. Jika *timer* waktu sudah habis maka waktu tercapai *YES buzzer* akan bunyi lampu *off hourmeter off*. Jika akan dilakukan penyinaran lagi maka kembali ke pengaturan *timer NO*, jika tidak proses selesai. Untuk diagram alir *sterilisator UV* berbasis *microcontroller ATmega 16*.

### 3.3. Diagram Mekanis



Gambar 3.3 Diagram Mekanis *sterilisator UV*

1. Tombol kotak merah di sebelah kiri adalah tombol *SWITCH*
2. Lampu merah adalah lampu indikator
3. Tombol kotak kecil ada 5 warna adalah :
  - Biru 1 = tombol 1 jam
  - Biru 2 = tombol 3 jam
  - Biru 3 = tombol 6 jam
  - Putih tombol *star*
  - Hitam tombol *reset*
4. Kotak persegi panjang warna hijau adalah *LCD display*
5. Kotak besar adalah *hourmeter*

Alat *sterilisator UV* mempunyai 4 lampu dan dilengkapi dengan *timer*, *delay timer* dan *hourmeter*. *Setting timer* terdiri atas 1 jam, 3 jam dan 6 jam. *Delay timer* nya selama 5 menit. Selain itu pada modul terdapat sklar tegangan PLN menuju catu daya, saat *On* ditekan (alat dalam keadaan terbuka) maka tegangan PLN akan masuk ke catu daya, kemudian masuk ke seluruh rangkaian dan *driver* sedangkan saat *Off* ditekan maka catu daya tidak akan mendapat tegangan. Dan *display* sebagai penampil lamanya waktu penyinaran tidak menyala.

### 3.4. Perakitan Rangkaian *Power supply*

#### 1. Alat

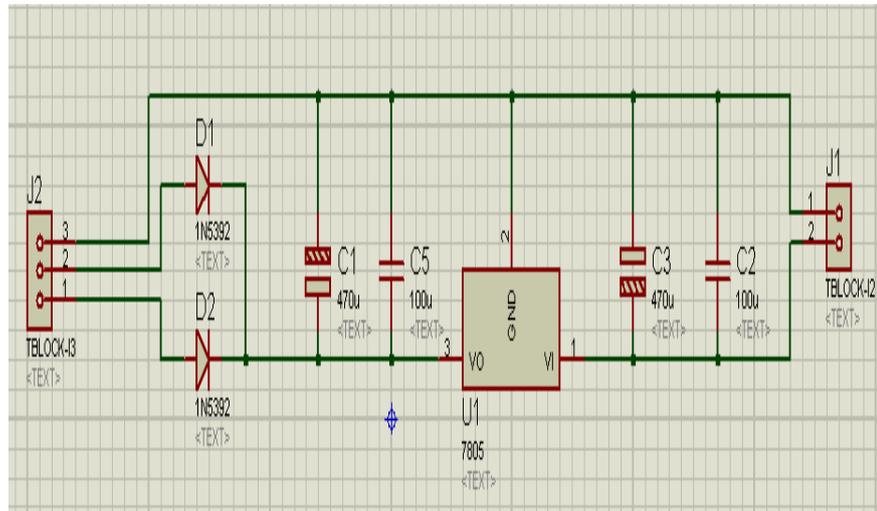
- a. Papan *PCB*
- b. Solder
- c. Timah / tinol
- d. Penyedot timah
- e. Bor *PCB*
- f. Pelarut *PCB*

#### 2. Bahan

- a. Travo 2 A *CT*
- b. Kapasitor 470  $\mu$ f (2)
- c. Kapasitor *non polar*  $10^4$  (2)
- d. *IC* regulator 7805
- e. Dioda 1N5392 (2)
- f. Pin sisir
- g. *T-blok*

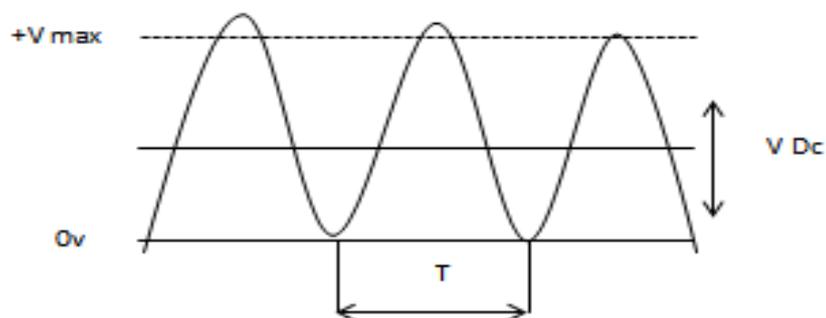
#### 3. Langkah Perakitan

- a. Rangkai skematik rangkaian *power supply* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*. Untuk gambar skematik rangkaian *power supply* pada aplikasi dapat dilihat pada Gambar 3.4.



Gambar 3.4 Skematik Rangkaian *Power Supply*

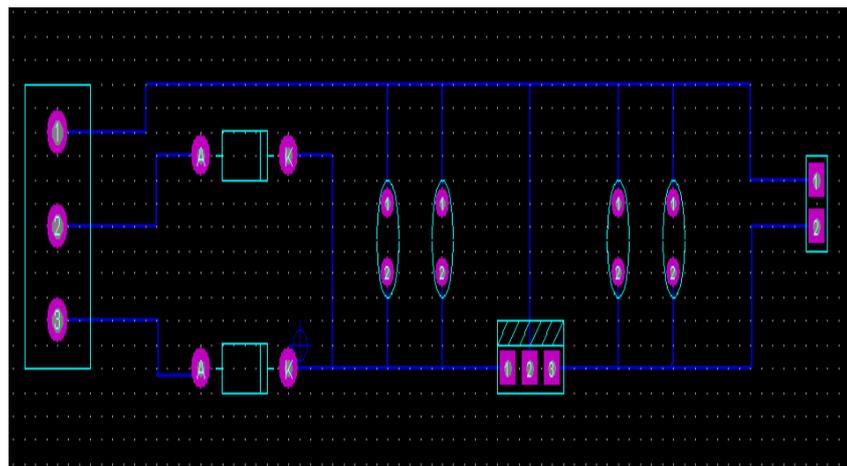
*Power supply* yang saya gunakan menggunakan dua dioda. Kedua dioda yang masing-masing berfungsi sebagai penyearah gelombang penuh dapat bekerja secara bergantian. Satu dioda menyearahkan siklus positif dari lilitan atas dan satu dioda lagi kemudian gentian menyearahkan siklus positif dari lilitan bawah dari transformator TC. (Prasetya, D.B., Iswanto & Sadad, R.T.A., 2010).



Gambar 3.5 *Resultant Output Waveform*

*Output* yang dihasilkan dari kedua dioda merupakan sinyal gelombang penuh yang lebih rapat dari pada setengah gelombang menyebabkan riak (*ripple*) yang ada pada output tegangan DC menjadi lebih kecil. Yang dihasilkan dari output penyearan gelombang penuh menjadi lebih halus dan lebih stabil dari pada penyearah setengah gelombang.

- b. Setelah skematik 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 3.6.



Gambar 3.6 *Lay out Power Supply*

- c. Rakit komponen yang dibutuhkan dengan menggunakan solder.
- d. Gambar *Power supply* untuk gambar *power supply* dapat dilihat pada Gambar 3.7.



Gambar 3.7 *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*. Adapun tegangan 5 *VDC* digunakan untuk untuk *supply* ke minimum sistem.

## 1.5. Perakitan Rangkaian Minimum Sistem

### 1. Alat

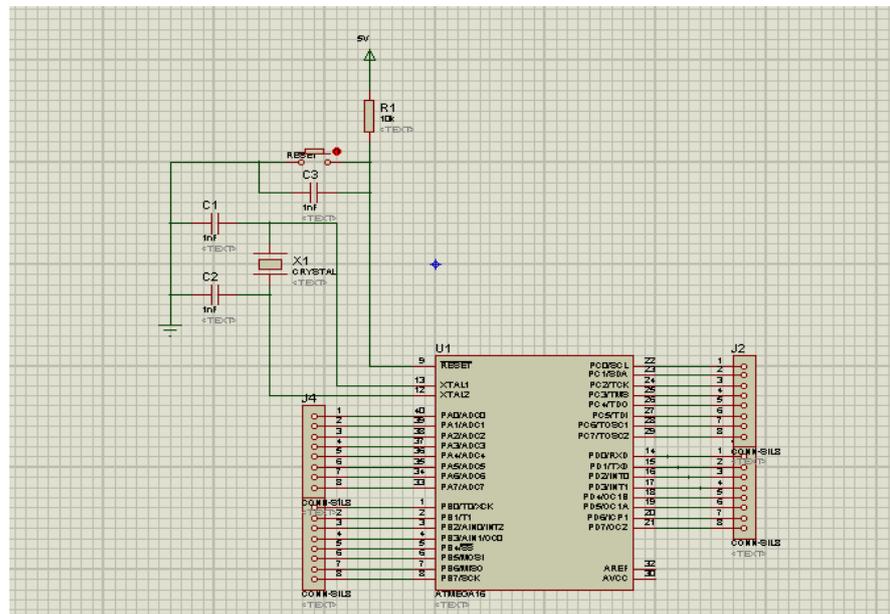
- a. Papan *PCB*
- b. Solder
- c. Timah / tinol
- d. Penyedot timah
- e. Bor *PCB*
- f. Pelarut *PCB*

### 2. Komponen

- a. *IC* ATmega 16 + soket
- b. *Crystal* 1 MHz
- c. Resistor *variable*
- d. Resistor 10 k
- e. Kapasitor 22 Pf
- f. Kapasitor 100 nF
- g. *Push button*
- h. Transistor BC547
- i. *IC* regulator 7805

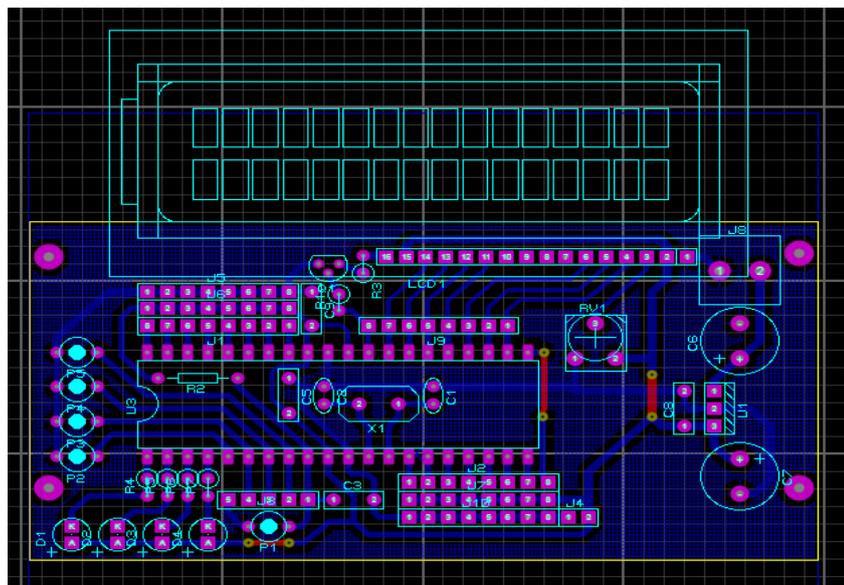
### 3. Langkah perakitan

- a. Rangkai skematik rangkaian minimum sistem dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*. Untuk Gambar skematik rangkaian minimum sistem pada aplikasi dapat dilihat pada Gambar 3.8.



Gambar 3.8 Skematik Minimum Sistem

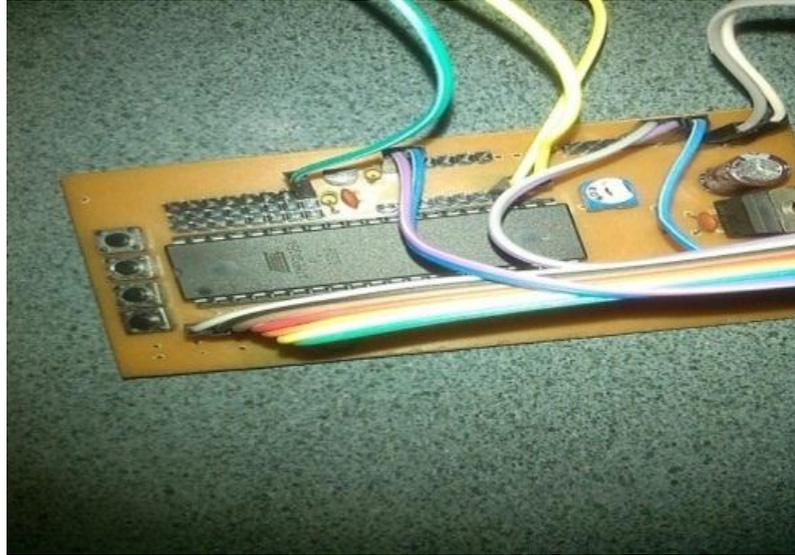
- b. Setelah skematik rangkaian jadi, tahap selanjutnya membuat *lay out* nya dan disablon ke papan *PCB*. Untuk gambar *lay out* minimum sistem pada papan *PCB* dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Layout* Rangkaian Minimum Sistem

- c. Rakit komponen yang dibutuhkan dengan menggunakan solder.

d. Gambar Minimum Sistem untuk Gambar minimum sistem dapat dilihat pada Gambar 3.10.

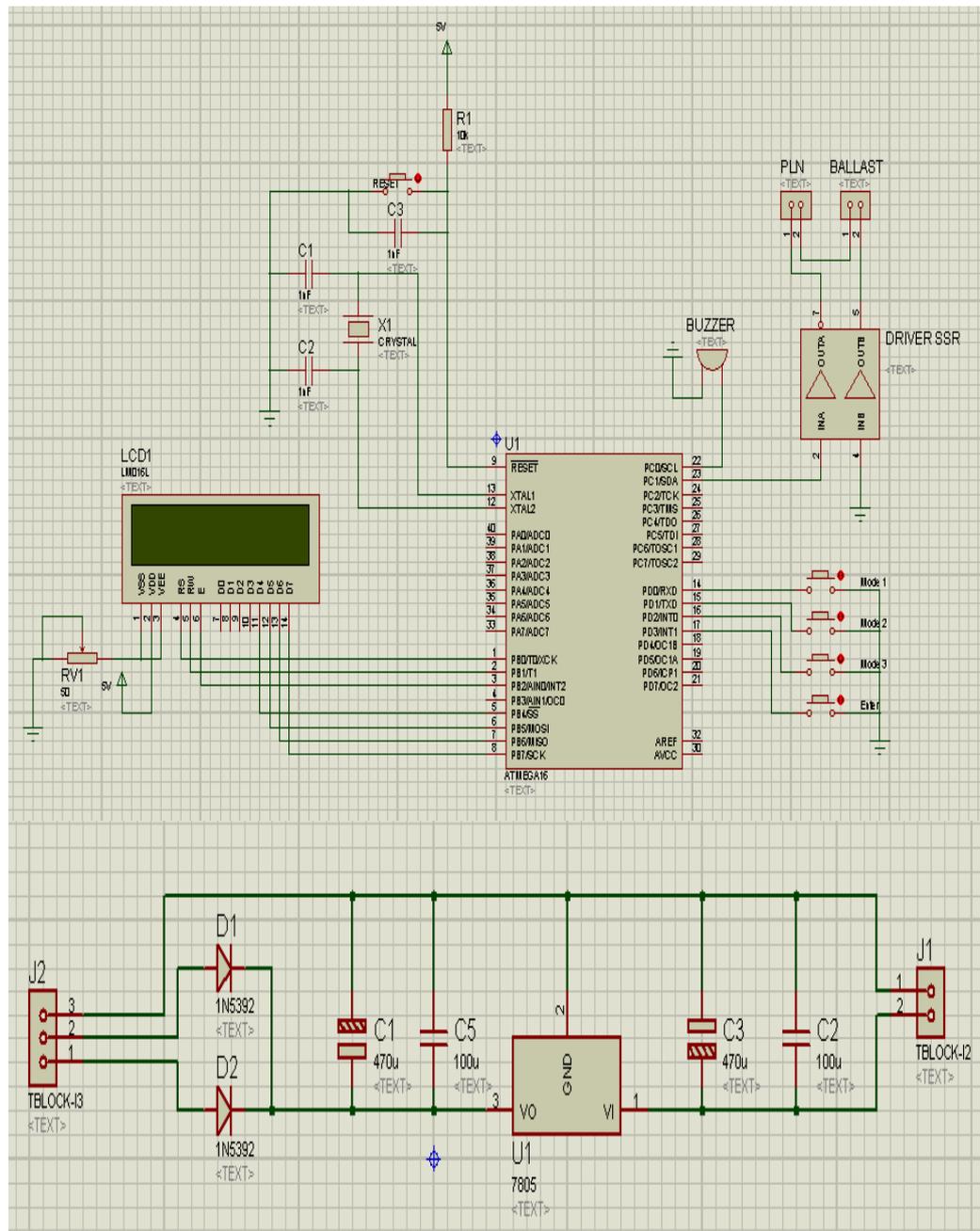


Gambar 3.10 Minimum Sistem

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 ADC sebagai pembaca tegangan dari program *timer* sebagai pengendali waktu pada modul.

## 1.6. Perakitan Rangkaian Skematik Keseluruhan

Untuk rangkaian skematik keseluruhan bisa dilihat pada Gambar 3.11 dibawah ini.



Gambar 3.11 Skematik Rangkaian Keseluruhan.

Prinsip kerja rangkaian skematik keseluruhan pada modul ini adalah dari *power supply* memberikan tegangan 5 VDC ke rangkaian minimum sistem. Rangkaian minimum sistem 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. *Port B* digunakan untuk keluaran LCD 16X2 yang berfungsi untuk menampilkan karakter maupun huruf. *Port C* pada rangkaian skematik keseluruhan pada modul ini digunakan untuk keluaran *buzzer* dan *driver SSR (solid state relay)* pada kaki 0 untuk keluaran *buzzer* sedangkan kaki 1 untuk keluaran *driver SSR (solid state relay)* Port D pada rangkaian skematik keseluruhan pada modul ini digunakan untuk tombol *push button* dari kaki 0 sampai kaki 3. Kaki 0 adalah tombol 1 jam, kaki 1 adalah tombol 3 jam, kaki 2 tombol 3 jam sedangkan kaki 3 sebagai tombol *enter/star*. Dan *reset* digunakan untuk mengulang program pada modul ini. (Iswanto, I., Wahyunggoro, O. & Cahyadi, A.I., 2016)

### **3.7. Pembuatan Program *Timer***

Untuk pembuatan program pada modul ini menggunakan aplikasi AVR dengan bahasa C. Program yang digunakan ialah program *timer* sebagai kontrol waktu untuk lama penyinaran. Berikut adalah Gambar 3.12 program yang digunakan:

```

1  /*****
2  This program was created by the
3  CodeWizardAVR V3.12 Advanced
4  Automatic Program Generator
5  © Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.
6  http://www.hpinfotech.com
7
8  Project :
9  Version :
10 Date   : 7/31/2016
11 Author :
12 Company :
13 Comments:
14
15
16 Chip type           : ATmega16
17 Program type        : Application
18 AVR Core Clock frequency: 1.000000 MHz
19 Memory model        : Small
20 External RAM size   : 0
21 Data Stack size     : 256
22 *****/
23
24 #include <mega16.h>
25
26 #include <util.h>

```

Gambar 3.12 Screenshot Program Timer

```

Notes a.c
23
24 #include <mega16.h>
25
26 #include <delay.h>
27 #include <stdlib.h>
28
29 // Alphanumeric LCD functions
30 #include <alcd.h>
31
32 // Declare your global variables here
33 bit timerrun=0,timernotend=1;
34 unsigned char settingtimer=3,temp[4],detik=59,jam=0,menit=59,menitprepare=5,detikprepare=0;
35 //unsigned int dataadc=0;
36 //float suhu;
37
38 // Timer1 overflow interrupt service routine
39 interrupt [TIM1_OVF] void timer1_ovf_isr(void)
40 {
41 // Reinitialize Timer1 value
42 TCNT1H=0xFC2F >> 8;
43 TCNT1L=0xFC2F & 0xff;
44 // Place your code here
45
46 if(timerrun==1)
47 {
48     if(detik==0)

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software andre\ a.c
Notes a.c
46 if(timerrun==1)
47 {
48     if(detik==0)
49     {
50         if(menit==0)
51         {
52             if(jam==0)
53             {
54                 timerrun=0;timernotend=0;
55             }else{
56                 jam--;menit=59;detik=59;
57             }
58         }else{
59             menit--;detik=59;
60         }
61     }else{
62         detik--;
63     }
64 }
65 }
66
67 // Voltage Reference: AVCC pin
68 #define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))
69
70 // Read the AD conversion result
71 //unsigned int read_adc(unsigned char adc_input)
72 //

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software\andre\c
Notes a.c
70 // Read the AD conversion result
71 //unsigned int read_adc(unsigned char adc_input)
72 //{
73 //ADMUX=adc_input | ADC_VREF_TYPE;
74 // Delay needed for the stabilization of the ADC input voltage
75 //delay_us(10);
76 // Start the AD conversion
77 //ADCSRA=(1<<ADSC);
78 // Wait for the AD conversion to complete
79 //while ((ADCSRA & (1<<ADIF))==0);
80 //ADCSRA=(1<<ADIF);
81 //return ADCW;
82 //}
83
84 void main(void)
85 {
86 // Declare your local variables here
87
88 // Input/Output Ports initialization
89 // Port A initialization
90 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
91 DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
92 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
93 PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
94
95 // Port B initialization

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software\andrea.c
Notes a.c
95 // Port B initialization
96 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
97 DDRB=(0<<DDDB7) | (0<<DDDB6) | (0<<DDDB5) | (0<<DDDB4) | (0<<DDDB3) | (0<<DDDB2) | (0<<DDDB1) | (0<<DDDB0);
98 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
99 PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
100
101 // Port C initialization
102 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=Out Bit0=Out
103 DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (1<<DDC1) | (1<<DDC0);
104 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=0 Bit0=0
105 PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
106
107 // Port D initialization
108 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
109 DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
110 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=P Bit2=P Bit1=P Bit0=P
111 PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (1<<PORTD3) | (1<<PORTD2) | (1<<PORTD1) | (1<<PORTD0);
112
113 // Timer/Counter 0 initialization
114 // Clock source: System Clock
115 // Clock value: Timer 0 Stopped
116 // Mode: Normal top=0xFF
117 // OC0 output: Disconnected
118 TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
119 TCNT0=0x00;
120 OCR0=0x00;

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software andre\ a.c
Notes a.c
119 TCNT0=0x00;
120 OCR0=0x00;
121
122 // Timer/Counter 1 initialization
123 // Clock source: System Clock
124 // Clock value: 0.977 kHz
125 // Mode: Normal top=0xFFFF
126 // OC1A output: Disconnected
127 // OC1B output: Disconnected
128 // Noise Canceler: Off
129 // Input Capture on Falling Edge
130 // Timer Period: 1.0004 s
131 // Timer1 Overflow Interrupt: On
132 // Input Capture Interrupt: Off
133 // Compare A Match Interrupt: Off
134 // Compare B Match Interrupt: Off
135 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
136 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (1<<CS12) | (0<<CS11) | (1<<CS10);
137 TCNT1H=0xFC;
138 TCNT1L=0x2F;
139 ICR1H=0x00;
140 ICR1L=0x00;
141 OCR1AH=0x00;
142 OCR1AL=0x00;
143 OCR1BH=0x00;
144 OCR1BL=0x00;

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software andre\ a.c
Notes a.c
144 OCR1BL=0x00;
145
146 // Timer/Counter 2 initialization
147 // Clock source: System Clock
148 // Clock value: Timer2 Stopped
149 // Mode: Normal top=0xFF
150 // OC2 output: Disconnected
151 ASSR=0<<AS2;
152 TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
153 TCNT2=0x00;
154 OCR2=0x00;
155
156 // Timer(s)/Counter(s) Interrupt(s) initialization
157 TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (1<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
158
159 // External Interrupt(s) initialization
160 // INT0: Off
161 // INT1: Off
162 // INT2: Off
163 MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
164 MCUCSR=(0<<ISC2);
165
166 // USART initialization
167 // USART disabled
168 UCSRB=(0<<RXDIE) | (0<<TXDIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCS22) | (0<<RXB8) | (0<<TXB8);
169

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software andre\la.c
Notes a.c
165
166 // USART initialization
167 // USART disabled
168 UCSRB=(0<<RXCEIE | 0<<TXCEIE | 0<<UDRIE | 0<<RXEN | 0<<TXEN | 0<<UCS22 | 0<<RXB8 | 0<<TXB8);
169
170 // Analog Comparator initialization
171 // Analog Comparator: Off
172 // The Analog Comparator's positive input is
173 // connected to the AIN0 pin
174 // The Analog Comparator's negative input is
175 // connected to the AIN1 pin
176 ACSR=(1<<ACD | 0<<ACBG | 0<<ACO | 0<<ACI | 0<<ACIE | 0<<ACIC | 0<<ACIS1 | 0<<ACIS0);
177
178 // ADC initialization
179 // ADC Clock frequency: 500.000 kHz
180 // ADC Voltage Reference: AVCC pin
181 // ADC Auto Trigger Source: ADC Stopped
182 //ADMUX=ADC_VREF_TYPE;
183 ADCSRA=(1<<ADEN | 0<<ADSC | 0<<ADATE | 0<<ADIF | 0<<ADIE | 0<<ADPS2 | 0<<ADPS1 | 1<<ADPS0);
184 SFIOR=(0<<ADTS2 | 0<<ADTS1 | 0<<ADTS0);
185
186 // SPI initialization
187 // SPI disabled
188 SPCR=(0<<SPIE | 0<<SPE | 0<<DORD | 0<<MSTR | 0<<CPOL | 0<<CPHA | 0<<SPR1 | 0<<SPR0);
189
190 // TWI initialization

```

Gambar 3.12 Screenshot Program *Timer* (lanjutan).

```

F:\software andre\la.c
Notes a.c
190 // TWI initialization
191 // TWI disabled
192 TWCR=(0<<TWEA | 0<<TWSTA | 0<<TWSTO | 0<<TWEN | 0<<TWIE);
193
194 // Alphanumeric LCD initialization
195 // Connections are specified in the
196 // Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
197 // RS - PORTB Bit 0
198 // RD - PORTB Bit 1
199 // EN - PORTB Bit 2
200 // D4 - PORTB Bit 4
201 // D5 - PORTB Bit 5
202 // D6 - PORTB Bit 6
203 // D7 - PORTB Bit 7
204 // Characters/line: 16
205 lcd_init(16);
206
207 // Global enable interrupts
208 #asm("sei")
209 lcd_gotoxy(0,0);
210 lcd_putsf("Welcome");
211 delay_ms(3000);
212
213 while(PIND.3==1)
214 {
215     if(PIND.0==0) //Jika PIND.0 ditekan

```

Gambar 3.12 Screenshot Program *Timer* (lanjutan).

```

F:\software andrea\ac
Notes ac
215 if (PIND.0==0) //Jika PIND.0 ditekan
216 {
217     settingtimer=1;delay_ms(500);
218 }else if (PIND.1==0){ //Jika PIND.1 ditekan
219     settingtimer=3;delay_ms(500);
220 }else if (PIND.2==0){ //Jika PIND.2 ditekan
221     settingtimer=6;delay_ms(500);
222 }
223 lcd_clear();
224 itoa(settingtimer,temp);lcd_puts("Setting: ");lcd_puts(temp);lcd_puts(" Jam"); //Tampilkan Settingan
225 delay_ms(200);
226 }
227 delay_ms(500);
228 while (menitprepare>0||detikprepare>0)
229 {
230     if (detikprepare==0)
231     {
232         if (menitprepare>0)
233         {
234             menitprepare--;detikprepare=59;
235         }
236     }else{
237         detikprepare--;
238     }
239     lcd_clear();lcd_puts("Persiapan");lcd_gotoxy(0,1);
240     lcd_puts("0");itoa(menitprepare,temp);lcd_puts(temp);lcd_puts(":");if (detikprepare<10){lcd_puts("0");}itoa(detikprepare,temp);lcd_puts(temp);
241     delay_ms(1000);

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software andrea\ac
Notes ac
240     lcd_puts("0");itoa(menitprepare,temp);lcd_puts(temp);lcd_puts(":");if (detikprepare<10){lcd_puts("0");}itoa(detikprepare,temp);lcd_puts(temp);
241     delay_ms(1000);
242 }
243
244 jam=settingtimer-1;
245 timerrun=1;
246 PORTC.1=1;
247
248 while (timernotend)
249 {
250     lcd_clear();
251     lcd_puts("Timer : ");
252     if (jam<10){lcd_puts("0");}itoa(jam,temp);lcd_puts(temp);lcd_puts(":");
253     if (menit<10){lcd_puts("0");}itoa(menit,temp);lcd_puts(temp);lcd_puts(":");
254     if (detik<10){lcd_puts("0");}itoa(detik,temp);lcd_puts(temp);
255     delay_ms(100);
256 }
257 lcd_clear();
258 lcd_puts("Selesai...");
259 PORTC.0=1;PORTC.1=0;
260
261 while (1)
262 {
263     |
264     // Place your code here
265 }
266 }

```

Gambar 3.12 Screenshot Program Timer (lanjutan).

```

F:\software andre\va.c
Notes a.c
243
244 jam=settingtimer-1;
245 timerrun=1;
246 PORTC.1=1;
247
248 while(timernotend)
249 {
250     lcd_clear();
251     lcd_puts("Timer : ");
252     if(jam<10){lcd_puts("0");}itoa(jam,temp);lcd_puts(temp);lcd_puts(":");
253     if(minit<10){lcd_puts("0");}itoa(minit,temp);lcd_puts(temp);lcd_puts(":");
254     if(detik<10){lcd_puts("0");}itoa(detik,temp);lcd_puts(temp);
255     delay_ms(100);
256 }
257 lcd_clear();
258 lcd_puts("Selesai...");
259 PORTC.0=1;PORTC.1=0;
260
261 while (1)
262 {
263
264     // Place your code here
265 }
266
267

```

Gambar 3.12 Screenshot Program *Timer* (lanjutan).

### 3.8. Perancangan Pengujian

Pada analisa rancangan ada parameter yang akan diujikan yaitu *timer* apakah rancangan sudah sesuai dengan kondisi yang diinginkan atau belum. Pengujian direncanakan akan dilakukan pengambilan data setiap 1 jam dengan 30 kali pengujian, 3 jam 10 kali pengujian dan 6 jam 10 pengujian. Pada modul ini terdapat parameter yang akan diuji yaitu lama waktu penyinaran.

*Timer/counter* adalah fasilitas dari ATMega16 yang digunakan untuk perhitungan pewaktuan. Pengujian *timer* ini bertujuan untuk memastikan bahwa *timer* sudah berfungsi dengan baik. Fungsi dari *timer* sendiri yaitu untuk mengatur lamanya waktu yang yang akan digunakan

dalam proses penyinaran. Pengujian *timer* dilakukan dengan cara melakukan perbandingan dengan *stopwatch*. Setiap 1 jam, 3 jam dan 6 jam dilihat data jam dan menit apakah sama dengan *stopwatch*. Pengujian *timer* dilakukan sebanyak masing-masing 1 jam 30 kali pengujian sedangkan 3 jam dan 6 jam masing-masing 10 kali pengujian.

### **3.9. Jenis Penelitian**

Jenis penelitian yang penulis gunakan adalah jenis penelitian eksperimental, artinya meneliti, mencari, menjelaskan, dan membuat suatu instrument dimana instrument ini dapat langsung dipergunakan oleh pengguna. *Variabel* yang diteliti dan diamati pada alat bantu *sterilisator UV* berbasis *microcontroller* ATmega 16 ini adalah lama waktu penyinaran pada lampu *UV*.

### **3.10. Variabel Penelitian**

#### **1. Variabel Bebas**

Sebagai *variabel* bebas adalah objek (ruangan) dengan udara yang di sterilkan dari bakteri atau kuman.

#### **2. Variabel Tergantung**

Sebagai *variabel* tergantung pada alat ini adalah lama waktu penyinaran.

### 3. Variabel Terkendali

Sebagai *variabel* terkendali yaitu lama waktu penyinaran pada lampu UV.

#### 3.11. Definisi Oprasional

Dalam kegiatan operasionalnya, *varaiabel-variabel* yang digunakan dalam perencanaan pembuatan modul, baik *variabel* terkendali, tergantung dan bebas memiliki fungsi-fungsi antara lain :

- Lampu UV digunakan sebagai *sterilisasi*.

#### 3.12. Sistematika Pengukuran

##### 1. Rata-rata

Rata-rata adalah nilai atau hasil pembagian dari jumlah data yang diambil atau diukur dengan banyaknya pengambilan data atau banyaknya pengukuran.

$$\boxed{\bar{x} = \frac{\sum x_i}{n}} \dots\dots\dots (3.1)$$

Dengan :

$\sum x_i$  : Jumlah X sebanyak i

n : Banyak data

$\bar{x}$  : Rata-rata

## 2. Simpangan *Error* %

Simpangan adalah selisih dari rata-rata nilai harga yang dikehendaki dengan nilai yang diukur. Berikut rumus dari simpangan:

$$\boxed{\text{Simpangan} = x - \bar{x}} \dots\dots\dots (3.2)$$

Dengan :

X : Data x

$\bar{x}$  : Rata-rata

## 3. *Presentase Error* (%)

*Error* (kesalahan) adalah selisih antara *mean* terhadap masing-masing data. Rumus *error* adalah :

$$\boxed{\text{Error (\%)} = \frac{x - \bar{x}}{x} \times 100\%} \dots\dots\dots (3.3)$$

Dengan :

*Error* : Besaran simpangan / nilai *error* dalam%

X : Data x

$\bar{x}$  : Rata-rata

## 4. *Standart deviasi* (SD)

*Standart deviasi* adalah suatu nilai yang menunjukkan tingkat (derajat) variasi kelompok data atau ukuran *standart* penyimpangan dari *meannya*. Rumus *standart deviasi* (SD) adalah :

$$SD = \sqrt{\frac{\sum_{i=1}^n (x - \bar{x})^2}{n-1}} \dots\dots\dots (3.4)$$

Dengan :

SD : *Standart devisiasi*

X : Data x

$\bar{x}$  : Rata-rata

n : Banyak data

### 5. Ketidakpastian ( $U_a$ )

Ketidakpastian adalah kesangsian yang muncul pada tiap hasil.

Atau pengukuran biasa disebut, sebagai kepresisian data satu dengan data

yang lain. Rumus dari ketidakpastian adalah sebagai berikut :

$$U_a = \frac{SD}{\sqrt{n}} \dots\dots\dots (3.5)$$

Dengan :

$U_a$  : Ketidakpastian

SD : *Standar Devisiasi*

n : banyak data