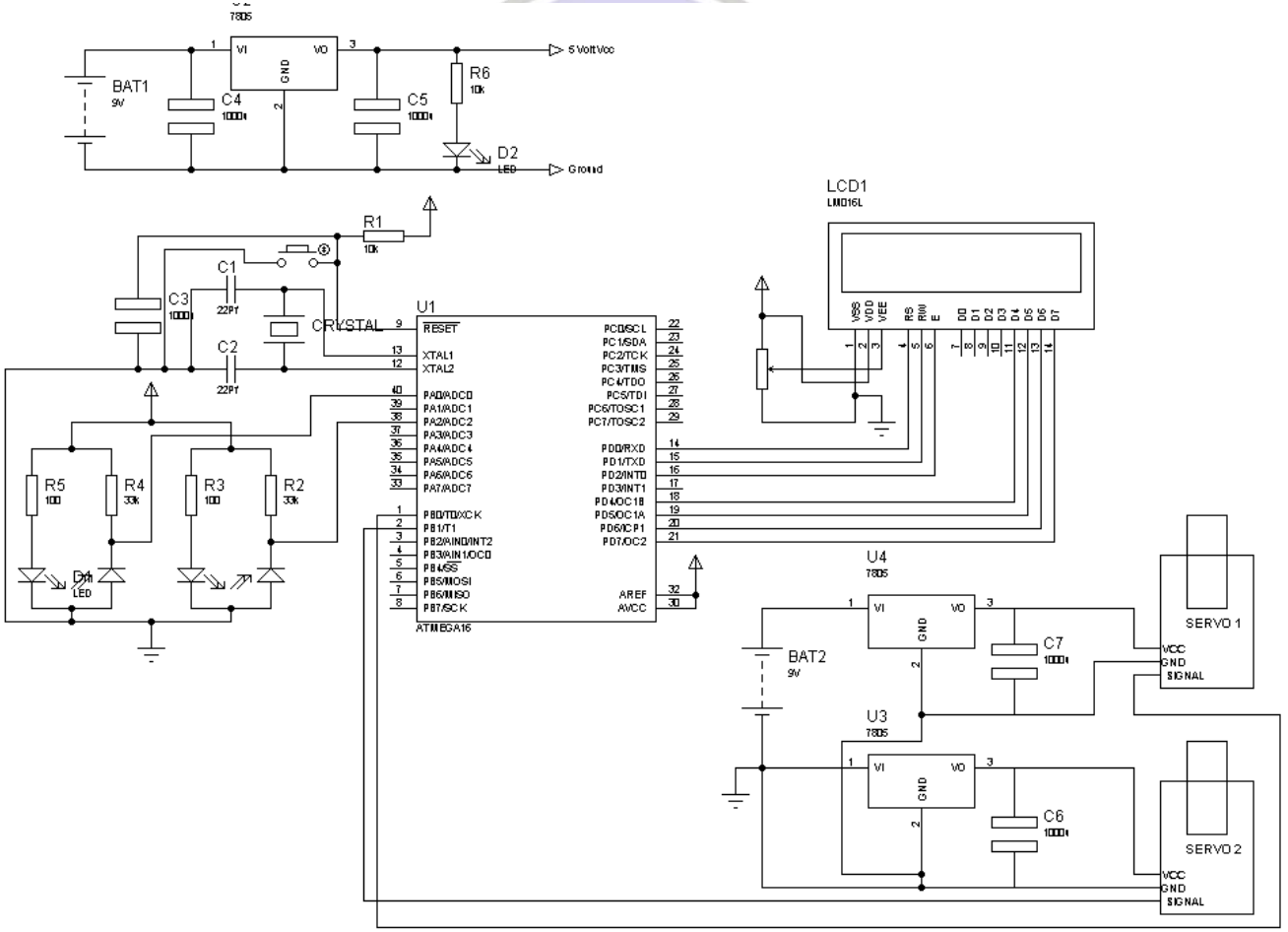
LAMPIRAN I
RANGKAIAN KESELURUHAN

RANGKAIAN KESELURUHAN



LAMPIRAN II
LISTING PROGRAM

LISTING PROGRAM

```
#include <mega16.h>
#include <stdio.h>

#define roda  PINA.0
#define kipas PINA.2
#define servo1 PORTB.0
#define servo2 PORTB.1
#define speedr PORTB.2
#define speedk PORTB.3

int i,count,count2;
int
pass,pass2,flag=1,go,pwmk,pwmr,kp
,kd,ki;
char t[16];

// Alphanumeric LCD Module
functions
#asm
   .equ __lcd_port=0x12 ;PORTD
#endasm
#include <lcd.h>

int kaizo,s1,s2;

// Timer 0 overflow interrupt service
routine
interrupt [TIM0_OVF] void
timer0_ovf_isr(void)
{
// Place your code here
kaizo++;
if(kaizo>2000)  kaizo=0;

if(s1>kaizo)    servo1=1;
else            servo1=0;

if(s2>kaizo)    servo2=1;
else            servo2=0;
/*
if(pwmk>kaizo)  speedk=1;
else            speedk=0;

if(pwmr>kaizo)  speedr=1;
else
*/          speedr=0;
}
#include <delay.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char
adc_input)
{
ADMUX=adc_input |
(ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization
of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to
complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}

 int
error,error_1,integ,deriv,setpoint=50,
val_error,plus,hitung,hitung2;
 int
konstanta,konstanta2,rpmroda,rpmki
pas;
// Declare your global variables here

void read_sensor()
{
 for(i=0;i<600;i++)
 {
 if(roda==1)
  {
  count=1;
  }
 if(roda==0 && count==1)
  {
  hitung+=1;
  count=0;
  }
 if(kipas==1)
  {
```

```
 count2=1;
 }
if(kipas==0 && count2==1)
 {
 hitung2+=2;
 count2=0;
 }
 delay_ms(1);
 }
lcd_gotoxy(0,0);
sprintf(t, "R %3d K
%3d",hitung,hitung2);
lcd_puts(t);
}

void kerja()
{
switch(go)
 {
 case 0:
 {
 read_sensor();
 if(hitung>=hitung2)
 {
 go=1;
 break;
 }
 else if(hitung2>=hitung)
 {
 go=2;
 break;
 }
 }

 case 1:
 {
   konstanta=220/60;
   konstanta2=220/30;
   rpmroda=hitung-30;
   s1=40+(rpmroda/konstanta);
   s2=70+(rpmroda/konstanta2);

   if(s1>100)       s1=100;
   if(s1<40)   s1=40;
   if(s2>100)       s2=100;
   if(s2<70)   s2=70;
  lcd_gotoxy(0,1);
  sprintf(t,"S1 %3d S2 %3d",s1,s2);
  lcd_puts(t);
  hitung=0;
  hitung2=0;
  go=0;
  break;
  }

 case 2:
  {
   konstanta=220/60;
   konstanta2=220/30;
   rpmkipas=hitung2-30;
   s1=40+(rpmkipas/konstanta);
   s2=70+(rpmkipas/konstanta2);

   if(s1>100)       s1=100;
   if(s1<40)   s1=40;
   if(s2>100)       s2=100;
   if(s2<70)   s2=70;

  lcd_gotoxy(0,1);
  sprintf(t,"S1 %3d S2 %3d",s1,s2);
  lcd_puts(t);
  hitung=0;
  hitung2=0;
  go=0;
  break;
  }
 }
}

void main(void)
{
PORTA=0x00;
DDRA=0x00;

PORTB=0x00;
DDRB=0x0F;

PORTC=0x00;
DDRC=0x00;

PORTD=0x00;
DDRD=0x00;
```

```
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

MCUCR=0x00;
MCUCSR=0x00;

TIMSK=0x01;

Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

ADMUX=ADC_VREF_TYPE &
0xff;
ADCSRA=0x84;

// LCD module initialization
lcd_init(16);

// Global enable interrupts
#asm("sei")

lcd_gotoxy(0,0);
  lcd_putsf(" SALITY SYSTEM  ");
  delay_ms(500);
  lcd_gotoxy(0,1);
  lcd_putsf("INITIALISATION..");
  delay_ms(1000);
  lcd_gotoxy(0,1);
  lcd_putsf("  SYSTEM READY  ");
  delay_ms(500);
  lcd_clear();

while (1)
      {
      kerja();
      };
}
```

LAMPIRAN III

DATASHEET