# LAMPIRAN 1

# KODE PROGRAM

## A. PIRANTI LUNAK MIKROKONTROLER AT89S52

```
#include <at89x52.h>
#define XBUFLEN 18
#define RBUFLEN 18
//#define TBUFLEN 10
/* data bits */
sbit at 0xA0 mtr;
sbit at 0xA1 val1;
sbit at 0xA2 val2;//P2
sfr at 0x90  adc_dt;//P1
int Setpoint,eror,last_eror;//;,delta;
float output,Kp,Ki,integral,Kd,prop,d,deriv;//Maxout,Minout,
const char hexarray[] = {'0', '1', '2', '3', '4', '5', '6', '7',
                         '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
static unsigned char  xbuf[XBUFLEN], sbuf[RBUFLEN],rbuf[RBUFLEN];
static unsigned char rcnt, xcnt, rpos, xpos;
static unsigned char busy;

void ser_init (void)
{
    ES = 0;
  rcnt = xcnt = rpos = xpos = 0;  /* init buffers */
  busy = 0;
  SCON = 0x50;
  PCON |= 0x80;                 /* SMOD = 1; */
  TMOD &= 0x0f;                 /* use timer 1 */
  TMOD |= 0x20;
  TL1 = -6; TH1 = -6; TR1 = 1;    /* 9600bps with 11.059MHz crystal */
  ES = 1;
}

void ser_handler (void) interrupt 4
{
  if (RI) {
                RI = 0;
                if (rcnt < RBUFLEN)
                        rbuf [(rpos+rcnt++) % RBUFLEN] = SBUF;
  }
  if (TI) {
                TI = 0;
                if (busy = xcnt) {
                        xcnt--;
                        SBUF = xbuf [xpos++];
                        if (xpos >= XBUFLEN)
                                xpos = 0;
                }
  }
}

/* put a character to com */
void ser_putc (unsigned char c)
{
  while (xcnt >= XBUFLEN) /* wait for room in buffer */
                ;
  ES = 0;
  if (busy) {
                xbuf[(xpos+xcnt++) % XBUFLEN] = c;
```

```
  } else {
                    SBUF = c;
                    busy = 1;
  }
  ES = 1;
}


unsigned char ser_getc (void)
{
  unsigned char c;
  while (!rcnt)
                    ;
  ES = 0;
  rcnt--;
  c = rbuf [rpos++];
  if (rpos >= RBUFLEN)
                    rpos = 0;
  ES = 1;
  return (c);
}
/* put a string to com */
#pragma save
#pragma noinduction
void ser_puts (unsigned char *s)
{
  unsigned char c;
  while (c=*s++) {
                    if (c == '\n') ser_putc ('\r');
                    ser_putc (c);
  }
}
#pragma restore
void ser_gets (unsigned char *s, unsigned char len)
{
  unsigned char pos, c;
  pos = 0;
  while (pos <= len) {
                    c = ser_getc ();
                    if (c == '\r') continue;
    s[pos++] = c;
                    if (c == '\n') break;
  }
  s[pos] = '\0';
}

unsigned char ser_can_xmt (void)
{
  return XBUFLEN - xcnt;
}

unsigned char ser_can_rcv (void)
{
  return rcnt;
}


void delay(unsigned int waktu)
{
                    TF0 = 0;
                    TH0 = (char) ((~(waktu))>>8);
```

```
                    TL0 = (char) ((~(waktu)) & 0xFF);
                    TR0 = 1;
                    while (!TF0);
                    TR0 = 0;
}


void kontrol(void)
{

    eror=Setpoint-(int)adc_dt;
    prop=(Kp*eror);

    integral =integral+eror;
    integral=integral*Ki;

    d=(eror-last_eror);
    deriv=d*Kd;

    last_eror=eror;

    output=prop+integral+deriv;


    if(output>1){val1=0;mtr=1;}//
    if(output<0){val1=1;mtr=0;}//!mtr;}//if(output<Minout)


}

void kirim(int angka)
{
     char s[3];
    char dig0,dig1,dig2;
    dig2=angka/100;
    angka%=100;
    dig1=angka/10;
    dig0=angka%10;
                    s[0] = hexarray[dig2];
                    s[1] = hexarray[dig1];
                    s[2] = hexarray[dig0];
  ser_puts(s);
}
void cek()
  {
     int f;
     f=(int)adc_dt;
     kontrol();
     kirim((int)adc_dt);
     ser_puts("\n");
   }
void main() {

                    /* serial port init */
  char* pcmd;
  ser_init();
  TMOD &= 0xF0;                 // timer 0 init
  TMOD |= 0x01;
                    P2=0;
   Setpoint=100;
   Kp=10;Ki=0.1;Kd=1;
    integral=0;
   last_eror=0;
```

```
 EA   = 1;        //Aktipkan Sistem interupsi MCS51
                  for (;;)
{
   if (ser_can_rcv) {
    pcmd=&sbuf;
    ser_gets(sbuf, RBUFLEN);
    switch(*pcmd++) {
      case 'C':
                  cek();
        break;
      case 's':
                  val1=0;mtr=0;integral=0;last_eror=0;
        break;
      case 'r':
                  Setpoint=100;
         Kp=10;Ki=0.1;Kd=1;
        break;
      case 'i':                    // Konstanta integral
          switch(*pcmd++)//karakter kedua
                  {
                        case '0':
                            Ki=0;
                           switch(*pcmd++)//karakter kedua
                           {
                           case ',':
                                switch(*pcmd++)//karakter ketiga
                                {
                                case '1':
                                Ki=0.1;
                                break;
                                case '2':
                                Ki=0.2;
                                break;
                                case '3':
                                Ki=0.3;
                                break;
                                case '4':
                                Ki=0.4;
                                break;
                                case '5':
                                Ki=0.5;
                                break;
                                case '6':
                                Ki=0.6;
                                break;
                                case '7':
                                Ki=0.7;
                                break;
                                case '8':
                                Ki=0.8;
                                break;
                                case '9':
                                Ki=0.9;
                                break;
                                default:
                                break;
                                }
                           break;
                           default:
                           break;
                           }
```

```
                    break;
                    case '1':
                        Ki=10;
                        switch(*pcmd++)//karakter kedua
                        {
                        case '0':
                        Ki=100;
                        break;
                        default:
                        break;
                        }
                    break;
                    case '2':
                        Ki=20;
                    break;
                    case '3':
                        Ki=30;
                    break;
                    case '4':
                        Ki=40;
                    break;
                    case '5':
                        Ki=50;
                    break;
                    case '6':
                        Ki=60;
                    break;
                    case '7':
                        Ki=70;
                    break;
                    case '8':
                        Ki=80;
                    break;
                    case '9':
                        Ki=90;
                    break;   /* */
                    default:
                    break;
            }
      break;

case 'p':                                   // Konstanta proporsional
    switch(*pcmd++)//karakter kedua
        {
                 case '0':
                     Kp=0;
                    switch(*pcmd++)//karakter kedua
                    {
                    case ',':
                        switch(*pcmd++)//karakter ketiga
                        {
                        case '1':
                        Kp=0.1;
                        break;
                        default:
                        break;
                        }
                    break;
                    default:
                    break;
                    }
                break;
                case '1':
                    Kp=1;
```

```
                switch(*pcmd++)//karakter kedua
                {
                case '0':
                Kp=10;
                break;
                default:
                break;
                }
            break;
            case '2':
                Kp=2;
            break;
            case '3':
                Kp=3;
            break;
            case '4':
                Kp=4;
            break;
            case '5':
                Kp=5;
            break;
            case '6':
                Kp=6;
            break;
            case '7':
                Kp=7;
            break;
            case '8':
                Kp=8;
            break;
            case '9':
                Kp=9;
            break;
            default:
            break;
        }
    break;

case 'd':                                   // Konstanta derivatif
    switch(*pcmd++)//karakter kedua
        {
            case '0':
                Kd=0;
                switch(*pcmd++)//karakter kedua
                {
                case ',':
                    switch(*pcmd++)//karakter ketiga
                    {
                    case '1':
                    Kd=0.1;
                    break;
                    default:
                    break;
                    }
                break;
                default:
                break;
                }
            break;
            case '1':
                Kd=1;
```

```
                switch(*pcmd++)//karakter kedua
                {
                case '0':
                Kd=10;
                break;
                default:
                break;
                }
            break;
            case '2':
                Kd=2;
            break;
            case '3':
                Kd=3;
            break;
            case '4':
                Kd=4;
            break;
            case '5':
                Kd=5;
            break;
            case '6':
                Kd=6;
            break;
            case '7':
                Kd=7;
            break;
            case '8':
                Kd=8;
            break;
            case '9':
                Kd=9;
            break;
            default:
            break;
        }
  break;
case 'h':
   switch(*pcmd++)//karakter kedua
     {
     case '0':
        switch(*pcmd++)//karakter kedua
        {
            case '0':
                switch(*pcmd++)//karakter kedua
                {
                case '0':
                    Setpoint=0;
                break;
                case '1':
                    Setpoint=1;
                break;
                case '2':
                    Setpoint=2;
                break;
                case '3':
                    Setpoint=3;
                break;
                case '4':
                    Setpoint=4;
                break;
                case '5':
                    Setpoint=5;
                break;
```

```
    case '6':
        Setpoint=6;
    break;
    case '7':
        Setpoint=7;
    break;
    case '8':
        Setpoint=8;
    break;
    case '9':
        Setpoint=9;
    break;
    default:
    break;
    }
break;
case '1':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=10;
    break;
    case '1':
        Setpoint=11;
    break;
    case '2':
        Setpoint=12;
    break;
    case '3':
        Setpoint=13;
    break;
    case '4':
        Setpoint=14;
    break;
    case '5':
        Setpoint=15;
    break;
    case '6':
        Setpoint=16;
    break;
    case '7':
        Setpoint=17;
    break;
    case '8':
        Setpoint=18;
    break;
    case '9':
        Setpoint=19;
    break;
    default:
    break;
    }
break;
case '2':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=20;
    break;
    case '1':
        Setpoint=21;
```

```
        break;
        case '2':
            Setpoint=22;
        break;
        case '3':
            Setpoint=23;
        break;
        case '4':
            Setpoint=24;
        break;
        case '5':
            Setpoint=25;
        break;
        case '6':
            Setpoint=26;
        break;
        case '7':
            Setpoint=27;
        break;
        case '8':
            Setpoint=28;
        break;
        case '9':
            Setpoint=29;
        break;
        default:
        break;
        }
    break;
    case '3':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=30;
        break;
        case '1':
            Setpoint=31;
        break;
        case '2':
            Setpoint=32;
        break;
        case '3':
            Setpoint=33;
        break;
        case '4':
            Setpoint=34;
        break;
        case '5':
            Setpoint=35;
        break;
        case '6':
            Setpoint=36;
        break;
        case '7':
            Setpoint=37;
        break;
        case '8':
            Setpoint=38;
        break;
        case '9':
            Setpoint=39;
        break;
        default:
        break;
```

```
    }
break;
case '4':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=40;
    break;
    case '1':
        Setpoint=41;
    break;
    case '2':
        Setpoint=42;
    break;
    case '3':
        Setpoint=43;
    break;
    case '4':
        Setpoint=44;
    break;
    case '5':
        Setpoint=45;
    break;
    case '6':
        Setpoint=46;
    break;
    case '7':
        Setpoint=47;
    break;
    case '8':
        Setpoint=48;
    break;
    case '9':
        Setpoint=49;
    break;
    default:
    break;
    }
break;
case '5':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=50;
    break;
    case '1':
        Setpoint=51;
    break;
    case '2':
        Setpoint=52;
    break;
    case '3':
        Setpoint=53;
    break;
    case '4':
        Setpoint=54;
    break;
    case '5':
        Setpoint=55;
    break;
    case '6':
```

```
            Setpoint=56;
        break;
        case '7':
            Setpoint=57;
        break;
        case '8':
            Setpoint=58;
        break;
        case '9':
            Setpoint=59;
        break;
        default:
        break;
        }
    break;
    case '6':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=60;
        break;
        case '1':
            Setpoint=61;
        break;
        case '2':
            Setpoint=62;
        break;
        case '3':
            Setpoint=63;
        break;
        case '4':
            Setpoint=64;
        break;
        case '5':
            Setpoint=65;
        break;
        case '6':
            Setpoint=66;
        break;
        case '7':
            Setpoint=67;
        break;
        case '8':
            Setpoint=68;
        break;
        case '9':
            Setpoint=69;
        break;
        default:
        break;
        }
    break;
    case '7':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=70;
        break;
        case '1':
            Setpoint=71;
        break;
        case '2':
            Setpoint=72;
        break;
```

```
        case '3':
            Setpoint=73;
        break;
        case '4':
            Setpoint=74;
        break;
        case '5':
            Setpoint=75;
        break;
        case '6':
            Setpoint=76;
        break;
        case '7':
            Setpoint=77;
        break;
        case '8':
            Setpoint=78;
        break;
        case '9':
            Setpoint=79;
        break;
        default:
        break;
        }
    break;
    case '8':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=80;
        break;
        case '1':
            Setpoint=81;
        break;
        case '2':
            Setpoint=82;
        break;
        case '3':
            Setpoint=83;
        break;
        case '4':
            Setpoint=84;
        break;
        case '5':
            Setpoint=85;
        break;
        case '6':
            Setpoint=86;
        break;
        case '7':
            Setpoint=87;
        break;
        case '8':
            Setpoint=88;
        break;
        case '9':
            Setpoint=89;
        break;
        default:
        break;
        }
```

```
            break;
            case '9':
                switch(*pcmd++)//karakter kedua
                {
                case '0':
                    Setpoint=90;
                break;
                case '1':
                    Setpoint=91;
                break;
                case '2':
                    Setpoint=92;
                break;
                case '3':
                    Setpoint=93;
                break;
                case '4':
                    Setpoint=94;
                break;
                case '5':
                    Setpoint=95;
                break;
                case '6':
                    Setpoint=96;
                break;
                case '7':
                    Setpoint=97;
                break;
                case '8':
                    Setpoint=98;
                break;
                case '9':
                    Setpoint=99;
                break;
                default:
                break;
                }
            break;
            default:
            break;
            }
        break;


    case '1':
        switch(*pcmd++)//karakter kedua
        {
            case '0':
                switch(*pcmd++)//karakter kedua
                {
                case '0':
                    Setpoint=100;
                break;
                case '1':
                    Setpoint=101;
                break;
                case '2':
                    Setpoint=102;
                break;
                case '3':
                    Setpoint=103;
                break;
                case '4':
                    Setpoint=104;
```

```
    break;
    case '5':
        Setpoint=105;
    break;
    case '6':
        Setpoint=106;
    break;
    case '7':
        Setpoint=107;
    break;
    case '8':
        Setpoint=108;
    break;
    case '9':
        Setpoint=109;
    break;
    default:
    break;
    }
break;
case '1':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=110;
    break;
    case '1':
        Setpoint=111;
    break;
    case '2':
        Setpoint=112;
    break;
    case '3':
        Setpoint=113;
    break;
    case '4':
        Setpoint=114;
    break;
    case '5':
        Setpoint=115;
    break;
    case '6':
        Setpoint=116;
    break;
    case '7':
        Setpoint=117;
    break;
    case '8':
        Setpoint=118;
    break;
    case '9':
        Setpoint=119;
    break;
    default:
    break;
    }
break;
case '2':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
```

```
            Setpoint=120;
        break;
        case '1':
            Setpoint=121;
        break;
        case '2':
            Setpoint=122;
        break;
        case '3':
            Setpoint=123;
        break;
        case '4':
            Setpoint=124;
        break;
        case '5':
            Setpoint=125;
        break;
        case '6':
            Setpoint=126;
        break;
        case '7':
            Setpoint=127;
        break;
        case '8':
            Setpoint=128;
        break;
        case '9':
            Setpoint=129;
        break;
        default:
        break;
        }
    break;
    case '3':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=130;
        break;
        case '1':
            Setpoint=131;
        break;
        case '2':
            Setpoint=132;
        break;
        case '3':
            Setpoint=133;
        break;
        case '4':
            Setpoint=134;
        break;
        case '5':
            Setpoint=135;
        break;
        case '6':
            Setpoint=136;
        break;
        case '7':
            Setpoint=137;
        break;
        case '8':
            Setpoint=138;
        break;
        case '9':
```

```
            Setpoint=139;
        break;
        default:
        break;
        }
    break;
    case '4':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=140;
        break;
        case '1':
            Setpoint=141;
        break;
        case '2':
            Setpoint=142;
        break;
        case '3':
            Setpoint=143;
        break;
        case '4':
            Setpoint=144;
        break;
        case '5':
            Setpoint=145;
        break;
        case '6':
            Setpoint=146;
        break;
        case '7':
            Setpoint=147;
        break;
        case '8':
            Setpoint=148;
        break;
        case '9':
            Setpoint=149;
        break;
        default:
        break;
        }
    break;
    case '5':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=150;
        break;
        case '1':
            Setpoint=151;
        break;
        case '2':
            Setpoint=152;
        break;
        case '3':
            Setpoint=153;
        break;
        case '4':
            Setpoint=154;
        break;
```

```
    case '5':
        Setpoint=155;
    break;
    case '6':
        Setpoint=156;
    break;
    case '7':
        Setpoint=157;
    break;
    case '8':
        Setpoint=158;
    break;
    case '9':
        Setpoint=159;
    break;
    default:
    break;
    }
break;
case '6':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=160;
    break;
    case '1':
        Setpoint=161;
    break;
    case '2':
        Setpoint=162;
    break;
    case '3':
        Setpoint=163;
    break;
    case '4':
        Setpoint=164;
    break;
    case '5':
        Setpoint=165;
    break;
    case '6':
        Setpoint=166;
    break;
    case '7':
        Setpoint=167;
    break;
    case '8':
        Setpoint=168;
    break;
    case '9':
        Setpoint=169;
    break;
    default:
    break;
    }
break;
case '7':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=170;
    break;
    case '1':
        Setpoint=171;
```

```
    break;
    case '2':
        Setpoint=172;
    break;
    case '3':
        Setpoint=173;
    break;
    case '4':
        Setpoint=174;
    break;
    case '5':
        Setpoint=175;
    break;
    case '6':
        Setpoint=176;
    break;
    case '7':
        Setpoint=177;
    break;
    case '8':
        Setpoint=178;
    break;
    case '9':
        Setpoint=179;
    break;
    default:
    break;
    }
break;
case '8':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=180;
    break;
    case '1':
        Setpoint=181;
    break;
    case '2':
        Setpoint=182;
    break;
    case '3':
        Setpoint=183;
    break;
    case '4':
        Setpoint=184;
    break;
    case '5':
        Setpoint=185;
    break;
    case '6':
        Setpoint=186;
    break;
    case '7':
        Setpoint=187;
    break;
    case '8':
        Setpoint=188;
    break;
    case '9':
        Setpoint=189;
```

```
                break;
                default:
                break;
                }
            break;
            case '9':
                switch(*pcmd++)//karakter kedua
                {
                case '0':
                    Setpoint=190;
                break;
                case '1':
                    Setpoint=191;
                break;
                case '2':
                    Setpoint=192;
                break;
                case '3':
                    Setpoint=193;
                break;
                case '4':
                    Setpoint=194;
                break;
                case '5':
                    Setpoint=195;
                break;
                case '6':
                    Setpoint=196;
                break;
                case '7':
                    Setpoint=197;
                break;
                case '8':
                    Setpoint=198;
                break;
                case '9':
                    Setpoint=199;
                break;
                default:
                break;
                }
            break;
            default:
            break;
            }
        break;


    case '2':
        switch(*pcmd++)//karakter kedua
        {
            case '0':
                switch(*pcmd++)//karakter kedua
                {
                case '0':
                    Setpoint=200;
                break;
                case '1':
                    Setpoint=201;
                break;
                case '2':
                    Setpoint=202;
                break;
                case '3':
```

```
            Setpoint=203;
        break;
        case '4':
            Setpoint=204;
        break;
        case '5':
            Setpoint=205;
        break;
        case '6':
            Setpoint=206;
        break;
        case '7':
            Setpoint=207;
        break;
        case '8':
            Setpoint=208;
        break;
        case '9':
            Setpoint=209;
        break;
        default:
        break;
        }
    break;
    case '1':
        switch(*pcmd++)//karakter kedua
        {
        case '0':
            Setpoint=210;
        break;
        case '1':
            Setpoint=211;
        break;
        case '2':
            Setpoint=212;
        break;
        case '3':
            Setpoint=213;
        break;
        case '4':
            Setpoint=214;
        break;
        case '5':
            Setpoint=215;
        break;
        case '6':
            Setpoint=216;
        break;
        case '7':
            Setpoint=217;
        break;
        case '8':
            Setpoint=218;
        break;
        case '9':
            Setpoint=219;
        break;
        default:
        break;
        }
    break;
```

```
case '2':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=220;
    break;
    case '1':
        Setpoint=221;
    break;
    case '2':
        Setpoint=222;
    break;
    case '3':
        Setpoint=223;
    break;
    case '4':
        Setpoint=224;
    break;
    case '5':
        Setpoint=225;
    break;
    case '6':
        Setpoint=226;
    break;
    case '7':
        Setpoint=227;
    break;
    case '8':
        Setpoint=228;
    break;
    case '9':
        Setpoint=229;
    break;
    default:
    break;
    }
break;
case '3':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=230;
    break;
    case '1':
        Setpoint=231;
    break;
    case '2':
        Setpoint=232;
    break;
    case '3':
        Setpoint=233;
    break;
    case '4':
        Setpoint=234;
    break;
    case '5':
        Setpoint=235;
    break;
    case '6':
        Setpoint=236;
    break;
    case '7':
        Setpoint=237;
    break;
```

```
    case '8':
        Setpoint=238;
    break;
    case '9':
        Setpoint=239;
    break;
    default:
    break;
    }
break;
case '4':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=240;
    break;
    case '1':
        Setpoint=241;
    break;
    case '2':
        Setpoint=242;
    break;
    case '3':
        Setpoint=243;
    break;
    case '4':
        Setpoint=244;
    break;
    case '5':
        Setpoint=245;
    break;
    case '6':
        Setpoint=246;
    break;
    case '7':
        Setpoint=247;
    break;
    case '8':
        Setpoint=248;
    break;
    case '9':
        Setpoint=249;
    break;
    default:
    break;
    }
break;
case '5':
    switch(*pcmd++)//karakter kedua
    {
    case '0':
        Setpoint=250;
    break;
    case '1':
        Setpoint=251;
    break;
    case '2':
        Setpoint=252;
    break;
    case '3':
        Setpoint=253;
```

```
                    break;
                    case '4':
                        Setpoint=254;
                    break;
                    case '5':
                        Setpoint=255;
                    break;
                    default:
                    break;
                    }
                break;

                default:
                break;
                }
            break;

                break;
            default:
                break;
            }
        break;
    default:                                    // unknown command
        break;
    }
                        }
}
}
```

## B. PIRANTI LUNAK PENGENDALI

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Runtime;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Text.RegularExpressions;

namespace grafik
{
    public partial class Form1 : Form
    {
        private System.IO.Ports.SerialPort ser;
        string blank;//,blonk;
        int erol;
        int erolakil = 0;
        double pro,de,derv,output;
        double intg=0;//pSetpoint = 0;
        private double pSet = 0;
        private StripChart stripChart;

        public double set
        {
            get { return pSet; }
            set
            {
                pSet = value;
                if ((int)Math.Round(pSet) != trackBar2.Value)
                    trackBar2.Value = (int)Math.Round(pSet);
                setvel.Text = (trackBar2.Value).ToString();
                    if (trackBar2.Value < 100) { setvel.Text = "0" +
setvel.Text; }
                     if (trackBar2.Value < 10) { setvel.Text = "0" +
setvel.Text; }
            }
        }
        public Form1()
        {
            InitializeComponent();
            stripChart = new StripChart();
            string[] ports = System.IO.Ports.SerialPort.GetPortNames();
            toolStripComPort.DropDownItems.Clear();
            foreach (string s in ports)
            {
                toolStripComPort.DropDownItems.Add(s);
            }
              ser = new System.IO.Ports.SerialPort(toolStripComPort.Text,
9600);//19200);
            GetConfig();
            try
            {
                ser.Open();
                //ser.WriteLine("r");
                     ser.WriteLine("p" + toolStripKP.Text); ser.WriteLine
("\n");
                ser.WriteLine("h" + setvel.Text); ser.WriteLine("\n");
            }
            catch (Exception ex)
```

```
            {
                throw ex;
            }

        }
        private void GetConfig()
        {

                                                ser.DataReceived    +=    new
System.IO.Ports.SerialDataReceivedEventHandler(ser_DataReceived);

        }
                        private   void   ser_DataReceived(object   sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {


            blank = ser.ReadLine();

        }

        private void timer1_Tick(object sender, EventArgs e)
        {

            stripChart.addSample(Convert.ToDouble(setvel.Text));
            stripChart.makeline(Convert.ToDouble(blank));
            pictureBox1.Image = stripChart.bmp;

        }
        private void timer2_Tick(object sender, EventArgs e)
        {
            intg = 0;
            tingval.Text = blank;
            ser.WriteLine("C");
            erol = Convert.ToInt32(setvel.Text) - Convert.ToInt32(blank);
            erorl.Text = erol.ToString();
            pro = Convert.ToDouble(erol) * Convert.ToDouble(KP2.Text);
            propo.Text = pro.ToString();
                intg = (intg + Convert.ToDouble(erol)) * Convert.ToDouble
(KI2.Text);//intg * Convert.ToDouble(KI2.Text);
            integ.Text = intg.ToString();
            de = Convert.ToDouble(erol) - Convert.ToDouble(erolakil);
            derv = de * Convert.ToDouble(KD2.Text);
            deriv.Text = derv.ToString();
            erolakil = erol;
            output = pro + intg + derv;
            outputval.Text = output.ToString();
        }

        private void trackBar2_Scroll(object sender, EventArgs e)
        {
            set = trackBar2.Value;
        }


        private void START_Click(object sender, EventArgs e)
        {
            timer1.Enabled = !timer1.Enabled;
            timer2.Enabled = !timer2.Enabled;
            if ((timer1.Enabled)&&(timer2.Enabled))
            { START.Text = "Stop Process"; }
            else
            {   intg = 0;
                erolakil = 0;
                      START.Text  =  "Start  Process";  ser.WriteLine("s");
ser.WriteLine("\n");
            }
```

```
        }


        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            ser.WriteLine("h" + setvel.Text);
        }

             private void toolStripKP_DropDownItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {
            toolStripKP.Text =  e.ClickedItem.Text;
        }
        private void tutup()
        {
            try
            {

                if (ser.IsOpen)
                    ser.Close();

            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
        private void buka()
        {
              ser = new System.IO.Ports.SerialPort(toolStripComPort.Text,
9600);

            GetConfig();
            try
            {
                ser.Open();
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
         private void toolStripComPort_DropDownItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {

            if ((timer1.Enabled)&&(timer2.Enabled))
            {
            START.Text = "Stop Process";
            timer1.Enabled = !timer1.Enabled;
            timer2.Enabled = !timer2.Enabled;
            tutup();
            toolStripComPort.Text = e.ClickedItem.Text;
            ser.PortName = Convert.ToString(e.ClickedItem.Text);
            buka();
            timer1.Enabled = !timer1.Enabled;
            timer2.Enabled = !timer2.Enabled;
            }
        }

        private void tomKP_Click(object sender, EventArgs e)
        {
            ser.WriteLine("p" + toolStripKP.Text); ser.WriteLine("\n");
```

```
            KP2.Text = toolStripKP.Text;
        }

        private void tomKI_Click(object sender, EventArgs e)
        {
            ser.WriteLine("i" + toolStripKI.Text); ser.WriteLine("\n");
            KI2.Text = toolStripKI.Text;
        }

        private void tomKD_Click(object sender, EventArgs e)
        {
            ser.WriteLine("d" + toolStripKD.Text); ser.WriteLine("\n");
            KD2.Text = toolStripKD.Text;
        }

             private void toolStripKI_DropDownItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {
            toolStripKI.Text = e.ClickedItem.Text;
        }

             private void toolStripKD_DropDownItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {
            toolStripKD.Text = e.ClickedItem.Text;
        }


    }
    public class StripChart
    {
        private Bitmap pBmp;    // the chart bitmap
        private Queue<double> qSP;  //collection of Setpoints
        private Queue<double> qP;

         private int x = 0, y = 0; // used to define points added to line
array

        // define brushes
        private SolidBrush brSP;    // setpoint brush
        private SolidBrush brP;
        // define pens
        private Pen pSP;
        private Pen pP;

          public Bitmap bmp   // allow the bitmap to be accessed, but not
mutated publicly.
        {
            get { return pBmp; }
        }


        public StripChart()
        {
            // define the bitmap.
                                        pBmp   =   new   Bitmap(275,   258,
System.Drawing.Imaging.PixelFormat.Format24bppRgb);
            // instatiate the queues
            qSP = new Queue<double>(255);
            qP = new Queue<double>(255);

            // make brushes that are lightly transparent
            brSP = new SolidBrush(Color.Red);
            brP = new SolidBrush(Color.Black);

            // make pens that will be used to draw the lines
            pSP = new Pen(brSP, 2);
```

```
        pP = new Pen(brP, 1);


    }

    //enter sampled values into the associated queues
    public void addSample(double sp)//, double pv, double mv)
    {
        // limit the size
        if (qSP.Count > 254)
            qSP.Dequeue();

        // place values into the queues
        qSP.Enqueue(sp);

        // update the bitmap.
        buildChart();

    }
    public void makeline(double sp)//, double pv, double mv)
    {
        // limit the size
        if (qP.Count > 254)
            qP.Dequeue();

        // place values into the queues
        qP.Enqueue(sp);

        // update the bitmap.
        buildChart2();

    }

    // empty out the queues
    public void clearQueus()
    {
        qSP.Clear();
    }

    public void buildChart()
    {
        Graphics g = Graphics.FromImage(pBmp);

        // clear the bmp
        g.Clear(Color.White);

        // make point arrays for the lines that will be drawn
        Point[] pointSP = new Point[qSP.Count];

        for (x = 0; x < qSP.Count; x++)
        {

            foreach (double d in qSP)
            {
                y = 255 - (int)Math.Round(d);
                if (y > 255) y = 255;
                pointSP[x] = new Point(x, y);
                x++;
            }

            if (pointSP.Length > 1)
            {
                g.DrawLines(pSP, pointSP);
```

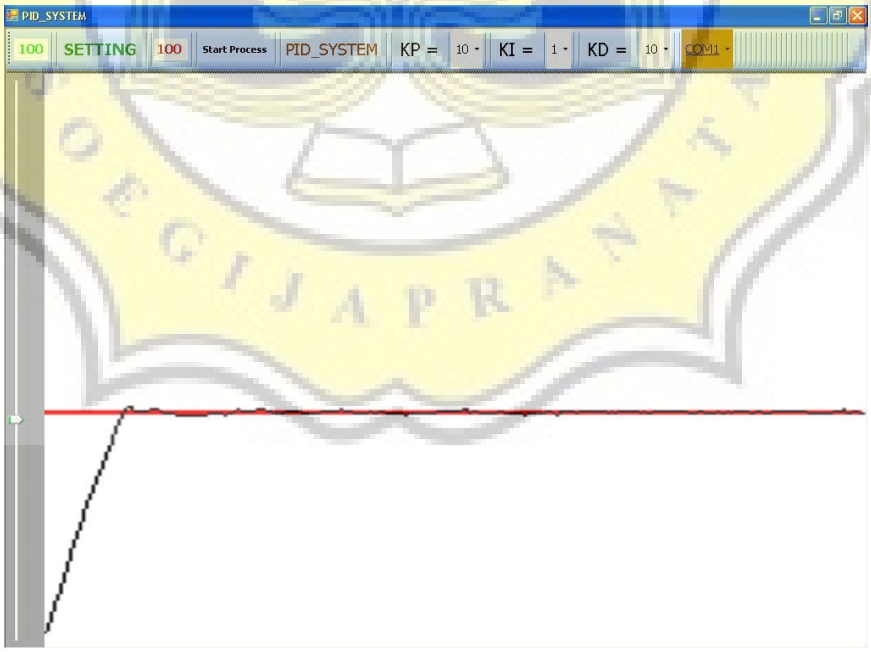
```
            }
            g.Dispose();
        }
    }
    public void buildChart2()
    {
        Graphics g = Graphics.FromImage(pBmp);
        // make point arrays for the lines that will be drawn
        Point[] pointPV = new Point[qP.Count];

        for (x = 0; x < qP.Count; x++)
        {

            foreach (double d in qP)
            {
                y = 255 - (int)Math.Round(d);
                pointPV[x] = new Point(x, y);
                x++;
            }

            if (pointPV.Length > 1)
            {
                g.DrawLines(pP, pointPV);

            }
            g.Dispose();
        }
    }
}
}
```




Form Program Antarmuka

# LAMPIRAN 2 DATA SHEET MAX232