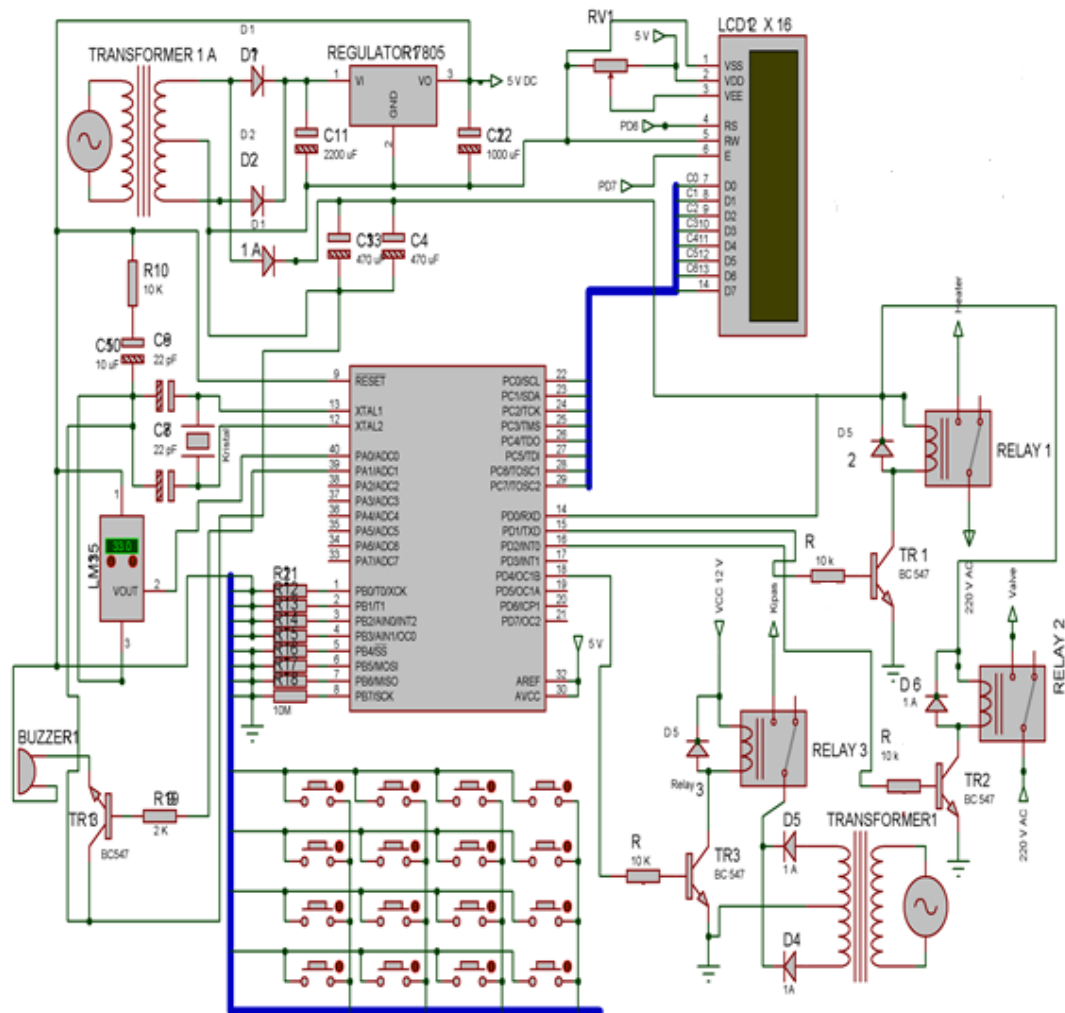


# Rangkaian Alat



## Kode Program

```

#define F_CPU 12000000UL

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>

unsigned char ad[255] EEMEM;

//0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF

//012345678901234567890123456789012345678901234567890123456789

char chr_01[54]   ={"* WATER BATH * press set key to start * WATER
BATH * p"};

char chr_02[16]   ={"PA:00 empty   "};

                               //0123456789ABCDEF

char chr_03[16]   ={"  Suhu  t  tPA  "};
char chr_04[16]   ={"< 000 C 00 00  >"};
char chr_05[8]    ={"Aborting"};
char chr_06[10]   ={"Insert Set"};

char chr_07[16]   ={"Set 00,0 C 00 00"};
char chr_08[16]   ={"Act 00,0 C 00.00"};

char chr_09[8]    ={"Finished"};
char chr_10[7]    ={"filled  "};
char chr_11[7]    ={"empty  "};

```

```

char chr_12[11]   ={"water empty"};
char chr_13[10]   ={"not permit"};

unsigned char _col[] ={0,2,3,4,8,9,11,12,15};

unsigned char _key[]={8,4,2,1};
                                //0 1 2 3 4 5 6 7 8
unsigned char __key[]={0,3,2,0,1,0,0,0,0};
unsigned char ___key[]={0,4,8,12};
                                //0 1 2 3 4 5 6 7 8 9 a b c d e
f
unsigned char ____key[]={12,3,2,1,13,6,5,4,14,9,8,7,15,11,0,10};

unsigned char d[4];
const unsigned int    ri=1000;
const unsigned char  ra=100;
const unsigned char  pu=10;

volatile unsigned char  flag[8],edit[9],cT,McT,col;
#define t                flag[0]
#define cmp0             flag[1]
#define blink            flag[2]
#define got              flag[3]
#define open1            flag[4]
#define buzz             flag[6]
#define waste            flag[7]

```

```
void LCD(unsigned char dat, char RS);

void init_LCD();

void wr_LCD(unsigned char line, unsigned char chr, unsigned char
count, unsigned char max, unsigned char add);

void BCD(unsigned int v);

unsigned char detect(unsigned char k)
{
    PORTB=k;
    asm("nop");
    asm("nop");
    k=PINB&0b11110000;
    k/=16;
    return(k);
}

ISR(TIMER1_COMPA_vect)
{t=1;}

ISR(TIMER0_COMP_vect)
{
    cT++;
    if (cT==McT)
        {cT=0;
        cmp0=1;}}

void wr_set()
{LCD(_col[col]+0xC0,0);
```

```

if (col==0)
    LCD('<',1);
else if (col==8)
    LCD('>',1);
else
    LCD(edit[col]+0x30,1);
}

void wr_T()
{
    LCD(d[2]+0x30,1);
    LCD(d[1]+0x30,1);
    LCD(',',1);
    LCD(d[0]+0x30,1);
}

void wr_PA(unsigned char PA)
{eeprom_write_byte(&ad[1],PA);}

//-----
-----

int main(void)
{unsigned char i,j,k,m,n,p,q,key,sec,mnt,hr,menu,Temp,rot,PA,cbuzz;
unsigned int c,T[20],set,cwaste;
float r;

_delay_ms(500);
DDRA=0b00000010; PORTA=0x00;
DDRB=0b00001111; PORTB=0x00;

```

```
DDRC=0b11111111; PORTC=0x00;
DDRD=0b11111110; PORTD=0x00;

_delay_ms(100);
init_LCD();
_delay_ms(100);
i=50;
//eeprom_write_byte(&ad[1],i);
//wr_PA(15);

for (i=0;i<20;i++)
    T[i]=0;

TIMSK=(1<<OCIE0);
TCCR0|=1<<WGM01|1<<CS00|1<<CS02;
OCR0=0xD8;
TCNT0=0x00;

TIMSK|=(1<<OCIE1A);
OCR1A=0xB71B;
TCCR1B=0;
//TCCR1B|=(1<<CS12);
open1=0; m=0; n=0; k=0; PA=0; t=0; cbuzz=0; cwaste=0;
sei();

//PORTD|=0b00001000;

goto _1st;

for (;)
```

```
{if (menu==0)
    {k=PIND&0b00000001;
    if (k==1)
        wr_LCD(0xC6,10,7,10,0);
    else
        wr_LCD(0xC6,11,7,10,0);
    }

for (i=0;i<4;i++)
    {k=_key[i];

    key=0;
    again:
    j=detect(k);
    if (j!=0)
        {key=j;
        goto again;}

    if (key!=0)
        {
        key=__key[key];
        key+=__key[i];
        key=___key[key];

        //BCD(key);
        //LCD(0xC3,0);
        //LCD(d[1]+0x30,1);
        //LCD(d[0]+0x30,1);
```

```

PORTA|=0b00000010;

_delay_ms(5);

PORTA&=0b11111101;

if (key<10)
    {if ((menu==1) & (col!=0) & (col!=8))
        edit[col]=key;
    }

else
    {switch(key)
        {case 12:
            if (menu==0)
                {menu=1;
                wr_LCD(0x80,3,16,16,0);
                wr_LCD(0xC0,4,16,16,0);
                LCD(0xC5,0);

                //OCR2=0;

                menu=1;

                col=0;

                edit[0]='<';

                edit[8]='>';

                for (m=1;m<8;m++)
                    {edit[m]=0;}

                LCD(0b11011111,1);

                m=eeprom_read_byte(&ad[1]);

```



```
        PA=m;
        BCD(m) ;
        LCD(0xCB,0) ;
        LCD(d[1]+0x30,1) ;
        LCD(d[0]+0x30,1) ;
        edit[6]=d[1];
        edit[7]=d[0];
        cT=0; McT=20;
    }
    break;
case 10:
    if (menu==1)
        {wr_set();
        if (col==0)
            col=8;
        else
            col--;
        }
    break;
case 11:
    if (menu==1)
        {wr_set();
        if (col==8)
            col=0;
        else
            col++;
        }
}
```

```

        break;

    case 15:
        if ((menu==1)&(col==0))
            {goto aborting;
            }
        else if ((menu==1)&(col==8))
            {m=PIND&0b00000001;
            if (m==1)

                {set=(edit[1]*100)+(edit[2]*10)+edit[3]; set*=10;

                p=(edit[4]*10)+edit[5];

                q=(edit[6]*10)+edit[7];

                if
                    {goto
                    comm_run;}

                }
            else

                {wr_LCD(0x80,12,11,16,0);

                wr_LCD(0xC0,13,10,16,0);

                _delay_ms(1500);
                goto aborting;}

            }
        else if (menu==2)

```

```
        {aborting:
        PORTD&=0b11111101;
        wr_LCD(0x80,5,8,16,0);
        wr_LCD(0xC0,0,16,16,0);
        goto _1st;
        }

    break;

case 13:
    if (open1==0)
        {PORTD|=0b00000100;
        open1=1;}
    else
        {PORTD&=0b11111011;
        open1=0;}

    break;

case 14:
    m=eeprom_read_byte(&ad[1]);
    if (m!=0)
        {n=PINB&0b00000001;
        if (n==1)
            {set=300;
            p=2;
            q=m;
            comm_run:
```

```
wr_LCD(0x80,6,10,16,0);

wr_LCD(0xC0,0,16,16,0);

                                _delay_ms(1000);

wr_LCD(0x80,7,16,16,0);

wr_LCD(0xC0,8,16,16,0);

LCD(0b11011111,1);

LCD(0b11011111,1);

LCD(0x88,0);

LCD(0xC8,0);

BCD(set);

LCD(0x84,0);

wr_T();

mnt=p;

sec=0;

BCD(p);

LCD(0x8B,0);

LCD(d[1]+0x30,1);

LCD(d[0]+0x30,1);

LCD(0xCB,0);

LCD(d[1]+0x30,1);

LCD(d[0]+0x30,1);

wr_PA(q);

BCD(q);

LCD(0x8E,0);

LCD(d[1]+0x30,1);

LCD(d[0]+0x30,1);
```

```
PORTD|=0b00000010;
menu=2;
got=0; buzz=0;
}
}
break;
}
}
}
}

//-----
-----

if (cmp0==1)
    {cmp0=0;
    if (menu==0)
        {rot++;
        if (rot==0x26)
            {rot=0;}
        wr_LCD(0x80,1,16,16,rot);
        }
    else if (menu==1)
        {
        if (blink==0)
            {blink=1;
            LCD(_col[col]+0xC0,0);
            LCD(' ',1);
            }
        }
```

```

        else
            {blink=0;
            wr_set();}
        }
    }

//-----
if ((t==1) & (menu==2))
    {t=0;
    if ((buzz==1) & (waste==0))
        {cbuzz++;
        PORTA^=2;
        if (cbuzz==6)
            {n=eeprom_read_byte(&ad[1]);
            n-=1;
            wr_PA(n);
            if (n==0)
                {waste=1; cwaste=0;
                PORTD|=0b00000100;}
            else
                goto _1st;
            }
        }
    else if ((buzz==1) & (waste==1))
        {cwaste++;
        if (cwaste==120)
            {PORTD&=0b11111011;
            _1st:

```

```
    wr_LCD(0x80,1,16,16,0);
    wr_LCD(0xC0,2,16,16,0);
    n=eeprom_read_byte(&ad[1]);
    BCD(n);
    LCD(0xC3,0);
    LCD(d[1]+0x30,1);
    LCD(d[0]+0x30,1);

    hr=0; mnt=0; sec=0;
    menu=0;
    rot=0; McT=40;
    Temp=0; set=0;
    TCCR1B=0;
    _delay_ms(500);
}
else
{if (sec==0)
    {if (mnt==0)
        {
            PORTD&=0b11110101;
            buzz=1; cbuzz=0;
        }
        else
            {mnt--;
            sec=59;}
    }
}
```

```

        else
            sec--;

            BCD(mnt);
            LCD(0xCB,0);
            LCD(d[1]+0x30,1);
            LCD(d[0]+0x30,1);

            BCD(sec);
            LCD(0xCE,0);
            LCD(d[1]+0x30,1);
            LCD(d[0]+0x30,1);
        }
    }

//-----
if ((menu==2) & (buzz==0) & (waste==0))
    {ADMUX=0;
    r=0;
    for (c=0;c<250;c++)
        {ADCSRA=(1<<ADEN) | (1<<ADSC) | (5<<ADPS0);
        while ((ADCSRA & 0x10) == 0);
        {ADCSRA=(1<<ADIF);
        r+=ADC;}
        }
    c=r/25;
    for (i=19;i>0;i--)
        T[i]=T[i-1];

```



```
T[0]=c;
r=0;
for (i=0;i<20;i++)
    r+=T[i];

r=r/20;

c=(r*500)/1023;
if ((menu==2)&(c>set))
    {PORTD|=0b00001000; _delay_ms(100);
    PORTD&=0b11111101;
    if (got==0)
        {got=1;
        TCCR1B=(1<<WGM12); TCCR1B|=(1<<CS12);}
    }
else if ((menu==2)&(c>(set-5))&(got==1))
    {PORTD&=0b11111101;
    }
else if ((menu==2)&(c<(set-5)))
    {PORTD|=0b00000010; _delay_ms(100);
    PORTD&=0b11110111;}

BCD(c);
if (menu==0)
    LCD(0xC5,0);
else
    LCD(0xC4,0);
wr_T();
```

```

        }
    }
}

//-----
void BCD(unsigned int v)
{
    d[3]=v/ri;
    d[2]=(v-((d[3]*ri)))/ra;
    d[1]=(v-((d[3]*ri)+(d[2]*ra)))/pu;
    d[0]= v-((d[3]*ri)+(d[2]*ra)+(d[1]*pu));}
void LCD(unsigned char dat, char RS)          //RS=1
{
    unsigned char i;
    PORTC=dat;
    if (RS==0)
        PORTD|=0b10000000;          //RS=0 send col
    else
        PORTD|=0b11000000;          //RS=1 send char

    for (i=0;i<70;i++)
        {asm("nop");}
    PORTD&=0b00111111;
    for (i=0;i<70;i++)
        {asm("nop");}}

void init_LCD()
{
    LCD(0x38,0);          //wr_inst(0x38);
    _delay_ms(50);
}

```

```

LCD(0x14,0);          //wr_inst(0x14);

_delay_ms(50);

LCD(0x0C,0);         //wr_inst(0x0C);

_delay_ms(50);

LCD(0x01,0);        //wr_inst(0x01);

_delay_ms(50);}

void wr_LCD(unsigned char line, unsigned char chr, unsigned char
count, unsigned char max, unsigned char add)

{unsigned char i;

LCD(line,0);

for (i=0;i<count;i++)

    {switch(chr)

        {case 0: LCD(' ',1);          break;

        case 1: LCD(chr_01[i+add],1);  break;

        case 2: LCD(chr_02[i+add],1);  break;

        case 3: LCD(chr_03[i+add],1);  break;

        case 4: LCD(chr_04[i+add],1);  break;

        case 5: LCD(chr_05[i+add],1);  break;

        case 6: LCD(chr_06[i+add],1);  break;

        case 7: LCD(chr_07[i+add],1);  break;

        case 8: LCD(chr_08[i+add],1);  break;

        case 9: LCD(chr_09[i+add],1);  break;

        case 10: LCD(chr_10[i+add],1);  break;

        case 11: LCD(chr_11[i+add],1);  break;

        case 12: LCD(chr_12[i+add],1);  break;

        case 13: LCD(chr_13[i+add],1);  break;

        }

    }
}

```

```
for (i=count;i<max;i++)
    {LCD(' ',1);}
}
```