LAMPIRAN A
LISTING PROGRAM

```
/*****************************************************
This program was produced by the
CodeWizardAVR V1.25.3 Standard
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 5/13/2008
Author  : F4CG
Company : F4CG
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 11.059200 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include <delay.h>
#include <exor.h>
#include <stdlib.h>


// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
 //global variabel
 unsigned long kdetik,rdetik,rmenit,rjam,rtanggal,rbulan,rtahun,cadangan,vv;
 unsigned char cdetik[10],cmenit[10],cjam[10],ctanggal[10],cbulan[10],ctahun[10];//,clt[10],crp[10];

// Timer 2 output compare interrupt service routine
interrupt [TIM2_COMP] void timer2_comp_isr(void)
{
// Place your code here
   TCNT2=0;// jika sudah match set awal lagi
   if(++kdetik==8920)
   {
        if(++rdetik==60)
        {rdetik=0;
        lcd_clear();
                if(++rmenit==60)
                {rmenit=0;
                lcd_clear();
                        if(++rjam==24)
                        {rjam=0;
                        lcd_clear();
                        rtanggal=rtanggal+1;
                                if(rtanggal==31 && rbulan==1)
                                { rtanggal=0;
                                rbulan=rbulan + 1;
                                }
                                if(rtanggal==28 && rbulan==2 && rtahun%4!=0)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==29 && rbulan==2 && rtahun %4==0)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==31 && rbulan==3)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==30 && rbulan==4)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==31 && rbulan==5)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==30 && rbulan==6)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==31 && rbulan==7)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==31 && rbulan==8)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==30 && rbulan==9)
                                {  rbulan=rbulan + 1;
                                }
                                if(rtanggal==31 && rbulan==10)
                                { rbulan=rbulan + 1;
                                }
                                if(rtanggal==30 && rbulan==11)
                                {  rbulan=rbulan + 1;
                                }
                                if(rtanggal==31 && rbulan==12)
                                { rtahun=rtahun + 1;
                                rbulan=0;
                                }
                        }
                }
        }

   kdetik=0;
   itoa(rdetik,cdetik);
   itoa(rmenit,cmenit);
   itoa(rjam,cjam);
   itoa(rtanggal,ctanggal);
   itoa(rbulan,cbulan);
   itoa(rtahun,ctahun);
  if(rtanggal<10)
   {
   Write_SC[0]=0x30;
   Write_SC[1]=ctanggal[0];
   }
   else
   {Write_SC[0]=ctanggal[0];
   Write_SC[1]=ctanggal[1];
   }
   if(rbulan<10)
   {
   Write_SC[2]=0x30;
   Write_SC[3]=cbulan[0];
   }
   else
   {
   Write_SC[2]=cbulan[0];
   Write_SC[3]=cbulan[1];
   }

   Write_SC[4]=ctahun[0];
   Write_SC[5]=ctahun[1];
   Write_SC[6]= ctahun[2];
   Write_SC[7]= ctahun[3];
   Write_SC[8]=0x00;
   Write_SC[9]=0x00;
   Write_SC[10]=0x00;
   Write_SC[11]=0x00;
   Write_SC[12]=0x00;
   Write_SC[13]=0x00;
   Write_SC[14]=0x00;
   Write_SC[15]=0x00;

   }
}
```

```
// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here
    unsigned char TEMP1,TEMP2,TEMP3,TEMP4[9];//,TEMP7,TEMP8;
   // unsigned char TEMP9,TEMP10,TEMP11,TEMP12,TEMP13,TEMP14,TEMP15,TEMP16;
    int i,x,y,tp;


 unsigned long lt,rp,tahun2,tanggal2,bulan2;

   unsigned char blok4[15],blok6[15];
   unsigned char input[8],harga[5];//,liter[5],rupiha[7];
  unsigned long a;//,b,c,d;

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x03;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
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=0x02;
TCNT2=0x00;
OCR2=0x96;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x80;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

//global enable interupts
#asm ("sei")

while (1)
      {
      // Place your code here

// program utama
      TEMP4[1]='0';
      TEMP4[2]='0';
      TEMP4[3]='0';
      TEMP4[4]='0';
      y=0;
seting:
      x=1;
 do
    {TEMP1=scan_keypad();

       if(TEMP1!=' '&&TEMP1!='D')
       {
       TEMP4[x]=TEMP1;
       x=x+1;
       delay_ms(300);
       };
    for(i=0;i<100;i++)
    { lcd_gotoxy(0,0);
     lcd_putsf("     ");
    lcd_gotoxy(5,0);
     lcd_putsf("HH:MM");
     lcd_gotoxy(5,1);
     lcd_putchar(TEMP4[1]);
     lcd_putchar(TEMP4[2]);
     lcd_putsf(":");
     lcd_putchar(TEMP4[3]);
     lcd_putchar(TEMP4[4]);
     };
   y=y+1;
  if(y==1000)
  {lcd_clear();
  y=0;};
  if(x==1)
    {for(i=0;i<300;i++)
    {
    lcd_gotoxy(5,1);
    lcd_putsf("_");
    };
    };
  if(x==2)
    {for(i=0;i<300;i++)
    {
    lcd_gotoxy(6,1);
    lcd_putsf("_");
    };
    };
  if(x==3)
  {for(i=0;i<300;i++)
    {
    lcd_gotoxy(8,1);
    lcd_putsf("_");
    };
    };
  if(x==4)
  {for(i=0;i<300;i++)
    {
    lcd_gotoxy(9,1);
    lcd_putsf("_");
    };
    };

  if(x==5)
  x=1;
}
while(TEMP1!='D');
lcd_clear();
rjam=TEMP4[1]-0x30;
rjam=rjam*10+TEMP4[2]-0x30;
rmenit=TEMP4[3]-0x30;
rmenit=rmenit*10+TEMP4[4]-0x30;
if(rjam>23 || rmenit>59)
{lcd_gotoxy(0,0);
 lcd_putsf("Input salah");
 lcd_gotoxy(0,1);
 lcd_putsf("masukan ulang");
 delay_ms(2000);
 lcd_clear();
 goto seting;
};
 TEMP4[1]='0';
 TEMP4[2]='0';
 TEMP4[3]='0';
 TEMP4[4]='0';
 TEMP4[5]='0';
 TEMP4[6]='0';
 TEMP4[7]='0';
 TEMP4[8]='0';
```

```
      y=0;
seting1:
      x=1;
 do
    {TEMP1=scan_keypad();

       if(TEMP1!=' '&&TEMP1!='D')
       {
       TEMP4[x]=TEMP1;
       x=x+1;
       delay_ms(300);
       };
         for(i=0;i<100;i++)
         { lcd_gotoxy(0,0);
          lcd_putsf("      ");
         lcd_gotoxy(3,0);
          lcd_putsf("DD/MM/YYYY");
          lcd_gotoxy(3,1);
          lcd_putchar(TEMP4[1]);
          lcd_putchar(TEMP4[2]);
          lcd_putsf("/");
          lcd_putchar(TEMP4[3]);
          lcd_putchar(TEMP4[4]);
          lcd_putsf("/");
          lcd_putchar(TEMP4[5]);
          lcd_putchar(TEMP4[6]);
          lcd_putchar(TEMP4[7]);
          lcd_putchar(TEMP4[8]);
          };

          y=y+1;
       if(y==1000)
       {lcd_clear();
       y=0;};

       if(x==1)
         {for(i=0;i<300;i++)
         {
         lcd_gotoxy(3,1);
         lcd_putsf("_");
         };
         };
       if(x==2)
         {for(i=0;i<300;i++)
         {
         lcd_gotoxy(4,1);
         lcd_putsf("_");
         };
         };
       if(x==3)
       {for(i=0;i<300;i++)
         {
         lcd_gotoxy(6,1);
         lcd_putsf("_");
         };
         };
       if(x==4)
       {for(i=0;i<300;i++)
         {
         lcd_gotoxy(7,1);
         lcd_putsf("_");
         };
         };
          if(x==5)
       {for(i=0;i<300;i++)
         {
         lcd_gotoxy(9,1);
         lcd_putsf("_");
         };
         };
          if(x==6)
```

```
   {for(i=0;i<300;i++)
    {
    lcd_gotoxy(10,1);
    lcd_putsf("_");
    };
    };
     if(x==7)
   {for(i=0;i<300;i++)
    {
    lcd_gotoxy(11,1);
    lcd_putsf("_");
    };
    };
     if(x==8)
   {for(i=0;i<300;i++)
    {
    lcd_gotoxy(12,1);
    lcd_putsf("_");
    };
    };

   if(x==9)
   x=1;
  }
  while(TEMP1!='D');
  lcd_clear();
  rtanggal=TEMP4[1]-0x30;
  rtanggal=rtanggal*10+TEMP4[2]-0x30;
  rbulan=TEMP4[3]-0x30;
  rbulan=rbulan*10+TEMP4[4]-0x30;
  cadangan=TEMP4[5]-0x30;
  rtahun=cadangan*1000;
  cadangan=TEMP4[6]-0x30;
  rtahun=rtahun+(cadangan*100);
  cadangan=TEMP4[7]-0x30;
  rtahun=rtahun+(cadangan*10);
  rtahun=rtahun+TEMP4[8]-0x30;

  if((rtanggal>31 && rbulan==1)||(rtanggal>28 && rbulan==2 && rtahun%4!=0)||
  (rtanggal>29 && rbulan==2 && rtahun %4==0) || (rtanggal>31 && rbulan==3)||
  (rtanggal>30 && rbulan==4)||(rtanggal>31 && rbulan==5)||(rtanggal>30 && rbulan==6)||
  (rtanggal>31 && rbulan==7)|| (rtanggal>31 && rbulan==8) || (rtanggal>30 && rbulan==9)||
  (rtanggal>31 && rbulan==10)|| (rtanggal>30 && rbulan==11)||(rtanggal>31 && rbulan==12)||
  (rbulan>12||rbulan==0||rtanggal==0)
  )
  {lcd_gotoxy(0,0);
   lcd_putsf("Input salah");
   lcd_gotoxy(0,1);
   lcd_putsf("masukan ulang");
   delay_ms(2000);
   lcd_clear();
   goto seting1;
  };

Start:
for(i=1;i<100;i++)
 { lcd_gotoxy(9,0);
 lcd_putsf(":");
 lcd_gotoxy(10,0);
 lcd_puts(cdetik);
 lcd_gotoxy(6,0);
 lcd_putsf(":");
 lcd_gotoxy(7,0);
 lcd_puts(cmenit);
 lcd_gotoxy(4,0);
 lcd_puts(cjam);
 lcd_gotoxy(1,1);
 lcd_puts(ctanggal);
 lcd_gotoxy(3,1);
 lcd_putsf("/");
 lcd_gotoxy(4,1);
```

```
   lcd_puts(cbulan);
   lcd_gotoxy(6,1);
   lcd_putsf("/");
   lcd_gotoxy(7,1);
   lcd_puts(ctahun);
   TEMP1=";
   TEMP2=";
    TEMP1=SCR_select();
     TEMP2=SCR_login(1);
    if(TEMP1==4 && TEMP2=='L')
     goto read;
    }
    lcd_clear();
for(i=1;i<100;i++)
    {
    lcd_gotoxy(0,0);
    lcd_putsf("selamat datang");
    lcd_gotoxy(0,1);
    lcd_putsf("di PT PERTAMINA");
     TEMP1=";
   TEMP2=";
    TEMP1=SCR_select();
     TEMP2=SCR_login(1);

     if(TEMP1==4 && TEMP2=='L')
     goto read;
    }
    lcd_clear();

     goto Start;

   read:
   TEMP1=SCR_read_block(0x04);
    blok4[0]=READ_SC[0];
     blok4[1]=READ_SC[1];
     blok4[2]=READ_SC[2];
     blok4[3]=READ_SC[3];
     blok4[4]=READ_SC[4];
     blok4[5]=READ_SC[5];
     blok4[6]=READ_SC[6];
     blok4[7]=READ_SC[7];
     blok4[8]=READ_SC[8];
     blok4[9]=READ_SC[9];
     blok4[10]=READ_SC[10];
     blok4[11]=READ_SC[11];
     blok4[12]=READ_SC[12];
    TEMP3=SCR_read_block(0x06);
     blok6[0]=READ_SC[0];
     blok6[1]=READ_SC[1];
     blok6[2]=READ_SC[2];
     blok6[3]=READ_SC[3];
     blok6[4]=READ_SC[4];
     blok6[5]=READ_SC[5];
     blok6[6]=READ_SC[6];
     blok6[7]=READ_SC[7];
    TEMP2=SCR_read_value(0x05);
     lcd_clear();
     if(TEMP1=='S'&&TEMP2=='S'&&TEMP3=='S')
     {
     goto read1;
     }
     else
     {lcd_gotoxy(0,0);
     lcd_putsf("maaf pembacaan ");
      lcd_gotoxy(0,1);
     lcd_putsf("gagal");
     delay_ms(2000);
     lcd_clear();
      lcd_gotoxy(0,0);
     lcd_putsf("silakan coba ");
      lcd_gotoxy(0,1);
     lcd_putsf("lagi");
     delay_ms(2000);
     lcd_clear();
     goto Start;
     };

read1:
   //cek tanggal dan update

   tanggal2=(kov_hextobin(blok6[0])*10)+kov_hextobin(blok6[1]);
   bulan2=(kov_hextobin(blok6[2])*10)+kov_hextobin(blok6[3]);

tahun2=(kov_hextobin(blok6[4])*1000)+(kov_hextobin(blok6[5])*100)+(kov_hextobin(blok6[6])*10)+kov_hextobin(blok
6[7]);
   if(rtanggal-tanggal2>=1 && rtanggal-tanggal2<32 &&rbulan-bulan2==0 && rtahun-tahun2==0)
   { //update tanggal dan write smartcard

  do
   {
     SCR_write_value(0x05,1000);
    }
    while(value!=1000);
    do
   {
     TEMP1=SCR_write_block(0x06);

    }
    while(TEMP1!='S');

    lcd_gotoxy(0,0);
    lcd_putsf("sedang membaca ");
    delay_ms(1000);
     lcd_clear();
   }

   else if(rtahun-tahun2>=1)
   {
   do
   {
     SCR_write_value(0x05,1000);
    }
    while(value!=1000);
    do
   {
     TEMP1=SCR_write_block(0x06);

    }
    while(TEMP1!='S');

    lcd_gotoxy(0,0);
    lcd_putsf("sedang membaca ");
    delay_ms(1000);
     lcd_clear();
   }

    else if(rbulan-bulan2>=1 && rbulan-bulan2<13 &&rtahun-tahun2==0)
   {
   do
   {
     SCR_write_value(0x05,1000);
    }
    while(value!=1000);
    do
   {
     TEMP1=SCR_write_block(0x06);

    }
    while(TEMP1!='S');

    lcd_gotoxy(0,0);
    lcd_putsf("sedang membaca ");
```

```
 delay_ms(1000);
  lcd_clear();
}

else
{
lcd_gotoxy(0,0);
 lcd_putsf(" tidak update ");
 delay_ms(1000);
  lcd_clear();
//tidak terjadi apa-apa
};
 // tampilkan data ke LCD
do{
  lcd_gotoxy(0,0);
  lcd_putchar(blok4[0]);
  lcd_gotoxy(1,0);
  lcd_putchar(blok4[1]);
  lcd_gotoxy(2,0);
  lcd_putchar(blok4[2]);
  lcd_gotoxy(3,0);
  lcd_putchar(blok4[3]);
  lcd_gotoxy(4,0);
  lcd_putchar(blok4[4]);
  lcd_gotoxy(5,0);
  lcd_putchar(blok4[5]);
  lcd_gotoxy(6,0);
  lcd_putchar(blok4[6]);
  lcd_gotoxy(7,0);
  lcd_putchar(blok4[7]);
  lcd_gotoxy(8,0);
  lcd_putchar(blok4[8]);


  lcd_gotoxy(0,1);
  lcd_putchar(blok4[9]);
  lcd_gotoxy(1,1);
  lcd_putchar(blok4[10]);
  lcd_gotoxy(2,1);
  lcd_putchar(blok4[11]);
  lcd_gotoxy(3,1);
  lcd_putchar(blok4[12]);

  lcd_putsf("/lt si ");

   lcd_gotoxy(11,1);
   vv=value;
   TEMP1=value/1000;
   lcd_putchar(TEMP1+0x30);
   lcd_gotoxy(12,1);
   value=(value % 1000);
   TEMP1=value/100;
   lcd_putchar(TEMP1+0x30);
   lcd_gotoxy(13,1);
   lcd_putchar(',');
   lcd_gotoxy(14,1);
   value=(value % 100);
   TEMP1=value/10;
   lcd_putchar(TEMP1+0x30);
   lcd_gotoxy(15,1);
   value=(value % 10);
   TEMP1=value;
   lcd_putchar(TEMP1+0x30);
   value=vv;
   TEMP2=scan_keypad();
   }
   while(TEMP2!='D');

      lcd_clear();
 delay_ms(500);
// melakukan transaksi
```

```
    x=0;
    TEMP1=";
      harga[0]=blok4[9];
      harga[1]=blok4[10];
      harga[2]=blok4[11];
      harga[3]=blok4[12];

a=(kov_hextobin(harga[0])*1000)+(kov_hextobin(harga[1])*100)+(kov_hextobin(harga[2])*10)+kov_hextobin(harga[3]);
     input[1]=' ';
     input[2]=' ';
     input[3]=' ';
     input[4]=' ';
     input[5]=' ';
     input[6]=' ';
  do
  {
   do
   {TEMP1=scan_keypad();
   delay_ms(100);
    lcd_gotoxy(x,0);
     lcd_putchar(TEMP1);
   }
   while(TEMP1==' ');

     lcd_gotoxy(x,0);
     lcd_putchar(TEMP1);
     delay_ms(400);
     input[x]=TEMP1;
     input1[x]=TEMP1;
     x=x+1;
   }
   while(TEMP1!='D');

      lt=0;

      if(input[0]=='D')
   { lcd_clear();
   goto Exit;
   }
    else if(input[0]=='A' && input[1]=='1' && input[2]=='D')
       { lt=200;
         rp=2*a;
         tp=1;
       }
      else if(input[0]=='A' && input[1]=='2' && input[2]=='D')
       {lt=1000;
         rp=10*a;
        tp=2;
       }
     else if(input[0]=='A' && input[1]=='3' && input[2]=='D')
       {
         rp=10000;
         lt=10000*100/a;
         tp=3;
       }
      else if(input[0]=='A' && input[1]=='4' && input[2]=='D')
       { rp=50000;
         lt=50000*100/a;
        tp=4;
       }
      else if(input[0]=='*' && input[1]!='A' && input[1]!='B'&& input[1]!='C'
        && input[2]!='A' && input[2]!='B'&& input[2]!='C'
       )
       {
           if(x==3 && input[2]=='D')
           { lt=kov_hextobin(input[1])*100;
            rp=lt*a/100;
           tp=51;
            }
           else if(x==4 && input[3]=='D')
           { lt=(kov_hextobin(input[1])*1000)+(kov_hextobin(input[2])*100);
```

```
             rp=lt*a/100;
            tp=52;
             }
            else
            {
            goto salah;
            };
         }
       else if(input[0]=='#'&& input[1]!='A' && input[1]!='B'&& input[1]!='C'
       && input[2]!='A' && input[2]!='B'&& input[2]!='C'
       && input[3]!='A' && input[3]!='B'&& input[3]!='C'
       && input[4]!='A' && input[4]!='B'&& input[4]!='C'
       && input[5]!='A' && input[5]!='B'&& input[5]!='C'
       && input[6]!='A' && input[6]!='B'&& input[6]!='C'

       )
        {
           if(x==6 && input[5]=='D')
           {
rp=(kov_hextobin(input[1])*1000)+(kov_hextobin(input[2])*100)+(kov_hextobin(input[3])*10)+kov_hextobin(input[4]);

             lt=rp*100/a;
             tp=61;
             }
            else if(x==7 && input[6]=='D')
            {
rp=(kov_hextobin(input[1])*10000)+(kov_hextobin(input[2])*1000)+(kov_hextobin(input[3])*100)+(kov_hextobin(input[
4])*10)+(kov_hextobin(input[5]));
             lt=rp*100/a;
           tp=62;

             }
           else if(x==8)
             {
rp=(kov_hextobin(input[1])*100000)+(kov_hextobin(input[2])*10000)+(kov_hextobin(input[3])*1000)+(kov_hextobin(in
put[4])*100)+(kov_hextobin(input[5])*10)+(kov_hextobin(input[6]));
              lt=rp*100/a;
             tp=63;
             }
                      else
              {
              goto salah;
               };
        }

        else
        {
      salah:
        lcd_gotoxy(0,0);
    lcd_putsf("maaf kode salah");
    delay_ms(2000);
    lcd_clear();
    goto Exit;
    };


     do
     {lcd_gotoxy(0,0);
    lcd_putsf("taruh kartu ");
     TEMP1=SCR_select();
    TEMP2=SCR_login(1);
     TEMP3=SCR_read_value(0x05);
    }
    while(TEMP1!=4 && TEMP2!='L' && TEMP3!='S');
        lcd_clear();
         if(value>=lt)
         {

           SCR_dec(0x05,lt);

  // tampilan harga dan liter
```

```
if(tp==1)
{lcd_gotoxy(0,0);
lcd_putsf("2 lt");
a=a*2;
lcd_gotoxy(0,1);
lcd_putsf("Rp ");
kov_longtolcd1(a); };


if(tp==2)
{lcd_gotoxy(0,0);
lcd_putsf("10 lt");
a=a*10;
lcd_gotoxy(0,1);
lcd_putsf("Rp ");
kov_longtolcd1(a); };

if(tp==3)
{ lcd_gotoxy(0,0);
a=100*10000/a;
kov_longtolcd2(a);

if(val3==2)
{lcd_putchar('0');
 lcd_putchar(',');
 lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
};

 if(val3==3)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(nilai1[2]+0x30);
};

 if(val3==4)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[2]+0x30);
 lcd_putchar(nilai1[3]+0x30);
};

lcd_putsf(" lt");
lcd_gotoxy(0,1);
lcd_putsf("Rp 10000");

};


if(tp==4)
{ lcd_gotoxy(0,0);
a=100*50000/a;
kov_longtolcd2(a);

if(val3==2)
{lcd_putchar('0');
 lcd_putchar(',');
 lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
};

 if(val3==3)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(nilai1[2]+0x30);
};
```

```
 if(val3==4)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[2]+0x30);
 lcd_putchar(nilai1[3]+0x30);
};

lcd_putsf(" lt");
lcd_gotoxy(0,1);
lcd_putsf("Rp 50000");

};


if(tp==51)
{lcd_gotoxy(0,0);
  lcd_putchar(input1[1]);
lcd_putsf(" lt");
a=a*kov_hextobin(input1[1]);
lcd_gotoxy(0,1);
lcd_putsf("Rp ");
kov_longtolcd1(a); };


if(tp==52)
{lcd_gotoxy(0,0);
  lcd_putchar(input1[1]);
  lcd_putchar(input1[2]);
lcd_putsf(" lt");
a=a*((kov_hextobin(input1[1])*10) + kov_hextobin(input1[2]) );
lcd_gotoxy(0,1);
lcd_putsf("Rp ");
kov_longtolcd1(a); };

if(tp==61)
{ lcd_gotoxy(0,0);
a=100*((kov_hextobin(input1[1])*1000)
+(kov_hextobin(input1[2])*100)+(kov_hextobin(input1[3])*10)+
(kov_hextobin(input1[4])*1))/a;
kov_longtolcd2(a);

if(val3==2)
{lcd_putchar('0');
 lcd_putchar(',');
 lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
};

 if(val3==3)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(nilai1[2]+0x30);
};

 if(val3==4)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[2]+0x30);
 lcd_putchar(nilai1[3]+0x30);
};

lcd_putsf(" lt");
lcd_gotoxy(0,1);
lcd_putsf("Rp ");
lcd_putchar(input1[1]);
 lcd_putchar(input1[2]);
  lcd_putchar(input1[3]);
   lcd_putchar(input1[4]);
```

```
};

if(tp==62)
{ lcd_gotoxy(0,0);
a=100*((kov_hextobin(input1[1])*10000)
+(kov_hextobin(input1[2])*1000)+(kov_hextobin(input1[3])*100)+
(kov_hextobin(input1[4])*10)+(kov_hextobin(input1[5])*1))/a;
kov_longtolcd2(a);

if(val3==2)
{lcd_putchar('0');
 lcd_putchar(',');
 lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
};

 if(val3==3)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(nilai1[2]+0x30);
};

 if(val3==4)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[2]+0x30);
 lcd_putchar(nilai1[3]+0x30);
};

lcd_putsf(" lt");
lcd_gotoxy(0,1);
lcd_putsf("Rp ");
lcd_putchar(input1[1]);
 lcd_putchar(input1[2]);
  lcd_putchar(input1[3]);
   lcd_putchar(input1[4]);
   lcd_putchar(input1[5]);
};

if(tp==63)
{ lcd_gotoxy(0,0);
a=100*((kov_hextobin(input1[1])*100000)
+(kov_hextobin(input1[2])*10000)+(kov_hextobin(input1[3])*1000)+
(kov_hextobin(input1[4])*100)+(kov_hextobin(input1[5])*10)+(kov_hextobin(input1[6])*1))/a;
kov_longtolcd2(a);

if(val3==2)
{lcd_putchar('0');
 lcd_putchar(',');
 lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
};

 if(val3==3)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(nilai1[2]+0x30);
};

 if(val3==4)
{lcd_putchar(nilai1[0]+0x30);
 lcd_putchar(nilai1[1]+0x30);
 lcd_putchar(',');
 lcd_putchar(nilai1[2]+0x30);
 lcd_putchar(nilai1[3]+0x30);
};

lcd_putsf(" lt");
```

```
  lcd_gotoxy(0,1);
  lcd_putsf("Rp ");
  lcd_putchar(input1[1]);
   lcd_putchar(input1[2]);
    lcd_putchar(input1[3]);
     lcd_putchar(input1[4]);
     lcd_putchar(input1[5]);
       lcd_putchar(input1[6]);
  };

     PORTB=0x01;
     delay_ms(lt*100);
     PORTB=0x00;

          }
          else
          { lcd_gotoxy(0,0);
     lcd_putsf("maaf transaksi ");
     lcd_gotoxy(0,1);
     lcd_putsf("gagal ");
     delay_ms(2000);
          };

          lcd_clear();

      Exit:
       goto Start;


#include <lcd.h>
#include <delay.h>
#include <mega16.h>
#include <stdio.h>

unsigned char TEMP;
char READ_SC[17];
char Write_SC[17];
long val;
long value,bin;
long val1,val2,val3;
unsigned char nilai1[8];
unsigned long rp1;
unsigned char exoreg;
unsigned char input1[8];

void SCR_reset(void)
{
putchar(0x02);
putchar(0x01);
putchar(0x01);
putchar(0x78);
putchar(0x78);
putchar(0x03);
}

char SCR_select(void)
{
putchar(0x02);
putchar(0x01);
putchar(0x01);
putchar(0x73);
putchar(0x73);
putchar(0x03);

TEMP = getchar();
TEMP = getchar();
TEMP = getchar();

if (TEMP == 0x01)
    {
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    return 1; //no card
    };

if (TEMP == 0x04)
    {
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    return 4; //standard milfare
    };


if (TEMP == 0x07)
    {
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    return 7; //ultralight milfare
    };

if (TEMP == 0x0A)
    {
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    return 0; //unknown
    };

    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
    return 'X'; //error
  }

  unsigned int exor(unsigned char xbyte,unsigned char ybyte)
{
unsigned char ybyte1;
unsigned char ybyte2;

ybyte1=(xbyte|ybyte);
ybyte2=(xbyte & ybyte);
ybyte2=~(ybyte2);
return((ybyte1&ybyte2));
}

char SCR_login(int sector)
{
putchar(0x02);
putchar(0x01);
putchar(0x09);
putchar(0x6C);
putchar(sector);/*sector*/
putchar(0xbb);
putchar(0xff);
putchar(0xff);
putchar(0xff);
putchar(0xff);
putchar(0xff);
putchar(0xff);

TEMP = exor(0x01,0x09);
TEMP = exor(TEMP,0x6C);
TEMP = exor(TEMP,sector);
TEMP = exor(TEMP,0xbb);
TEMP = exor(TEMP,0xff);
TEMP = exor(TEMP,0xff);
TEMP = exor(TEMP,0xff);
TEMP = exor(TEMP,0xff);
TEMP = exor(TEMP,0xff);
TEMP = exor(TEMP,0xff);

putchar(TEMP);
putchar(0x03);

   TEMP = getchar();
   TEMP = getchar();
   TEMP = getchar();
   TEMP = getchar();

if (TEMP == 'F')
   {
   TEMP = getchar();
   TEMP = getchar();
   return 'F'; //Fail
   };

if (TEMP == 'N')
   {
   TEMP = getchar();
   TEMP = getchar();
   return 'N'; //No tag
   };

if (TEMP == 'I')
   {
   TEMP = getchar();
   TEMP = getchar();
   return 'I'; //Invalid
   };

if (TEMP == 'E')
   {
   TEMP = getchar();
   TEMP = getchar();
   return 'I'; //Invalid
   };

if (TEMP == 'L')
   {
   TEMP = getchar();
   TEMP = getchar();
   return 'L'; //Invalid
   };

   TEMP = getchar();
   TEMP = getchar();
   return 'X';
   }

 char SCR_read_block(unsigned char block)
{
putchar(0x02);
putchar(0x01);
putchar(0x02);
putchar(0x72);
putchar(block);//block = (sec x 4) + block
putchar(exor(0x71,block));
putchar(0x03);

    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
if (TEMP == 0x01)
    {
    TEMP = getchar();

        if (TEMP == 'F')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'F'; //Fail
        };

        if (TEMP == 'N')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'N'; //No tag
        };

        if (TEMP == 'I')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'I'; //Invalid
        };
    };

if (TEMP == 0x10)
    {
    READ_SC[0] = getchar();
    READ_SC[1] = getchar();
    READ_SC[2] = getchar();
    READ_SC[3] = getchar();
    READ_SC[4] = getchar();
    READ_SC[5] = getchar();
    READ_SC[6] = getchar();
    READ_SC[7] = getchar();
    READ_SC[8] = getchar();
    READ_SC[9] = getchar();
    READ_SC[10] = getchar();
    READ_SC[11] = getchar();
    READ_SC[12] = getchar();
    READ_SC[13] = getchar();
    READ_SC[14] = getchar();
    //READ_SC[15] =
    TEMP = getchar();


    TEMP = getchar();
    TEMP = getchar();


    return 'S';
    };
                TEMP = getchar();
                TEMP = getchar();
                return 'X'; //Invalid
}

char SCR_write_block(unsigned char block)
{
putchar(0x02);
putchar(0x01);
putchar(0x12);
putchar(0x77);
putchar(block);//block = (sec x 4) + block

putchar(Write_SC[0]);
putchar(Write_SC[1]);
putchar(Write_SC[2]);
putchar(Write_SC[3]);
putchar(Write_SC[4]);
putchar(Write_SC[5]);
putchar(Write_SC[6]);
putchar(Write_SC[7]);
putchar(Write_SC[8]);
putchar(Write_SC[9]);
putchar(Write_SC[10]);
putchar(Write_SC[11]);
putchar(Write_SC[12]);
putchar(Write_SC[13]);
putchar(Write_SC[14]);
putchar(Write_SC[15]);

TEMP = exor(0x01,0x12);
TEMP = exor(TEMP,0x77);
TEMP = exor(TEMP,block);
TEMP = exor(TEMP,Write_SC[0]);
TEMP = exor(TEMP,Write_SC[1]);
TEMP = exor(TEMP,Write_SC[2]);
TEMP = exor(TEMP,Write_SC[3]);
TEMP = exor(TEMP,Write_SC[4]);
TEMP = exor(TEMP,Write_SC[5]);
TEMP = exor(TEMP,Write_SC[6]);
TEMP = exor(TEMP,Write_SC[7]);
TEMP = exor(TEMP,Write_SC[8]);
TEMP = exor(TEMP,Write_SC[9]);
TEMP = exor(TEMP,Write_SC[10]);
TEMP = exor(TEMP,Write_SC[11]);
TEMP = exor(TEMP,Write_SC[12]);
TEMP = exor(TEMP,Write_SC[13]);
TEMP = exor(TEMP,Write_SC[14]);
TEMP = exor(TEMP,Write_SC[15]);

putchar(TEMP);
putchar(0x03);

    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();

if (TEMP == 0x01)
    {
    TEMP = getchar();

        if (TEMP == 'F')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'F'; //Fail
        };

        if (TEMP == 'N')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'N'; //No tag
        };

        if (TEMP == 'I')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'I'; //Invalid
        };
    };

if (TEMP == 0x10)
    {
    READ_SC[0] = getchar();
    READ_SC[1] = getchar();
    READ_SC[2] = getchar();
    READ_SC[3] = getchar();
    READ_SC[4] = getchar();
    READ_SC[5] = getchar();
    READ_SC[6] = getchar();
    READ_SC[7] = getchar();
    READ_SC[8] = getchar();
    READ_SC[9] = getchar();
    READ_SC[10] = getchar();
    READ_SC[11] = getchar();
    READ_SC[12] = getchar();
    READ_SC[13] = getchar();
    READ_SC[14] = getchar();
    //READ_SC[15] =
    TEMP = getchar();


    TEMP = getchar();
    TEMP = getchar();


    return 'S';
    };
                TEMP = getchar();
                TEMP = getchar();
                return 'X'; //Invalid
}

unsigned long SCR_read_value(unsigned char block)
{
putchar(0x02);
putchar(0x01);
putchar(0x03);
putchar(0x72);
putchar(0x76);
putchar(block);//block = (sec x 4) + block
putchar(exor(0x06,block));
putchar(0x03);

    TEMP = getchar();
    TEMP = getchar();
    TEMP = getchar();
if (TEMP == 0x01)
    {
    TEMP = getchar();

        if (TEMP == 'F')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'F'; //Fail
        };

        if (TEMP == 'N')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'N'; //No tag
        };

        if (TEMP == 'I')
        {
                TEMP = getchar();
```

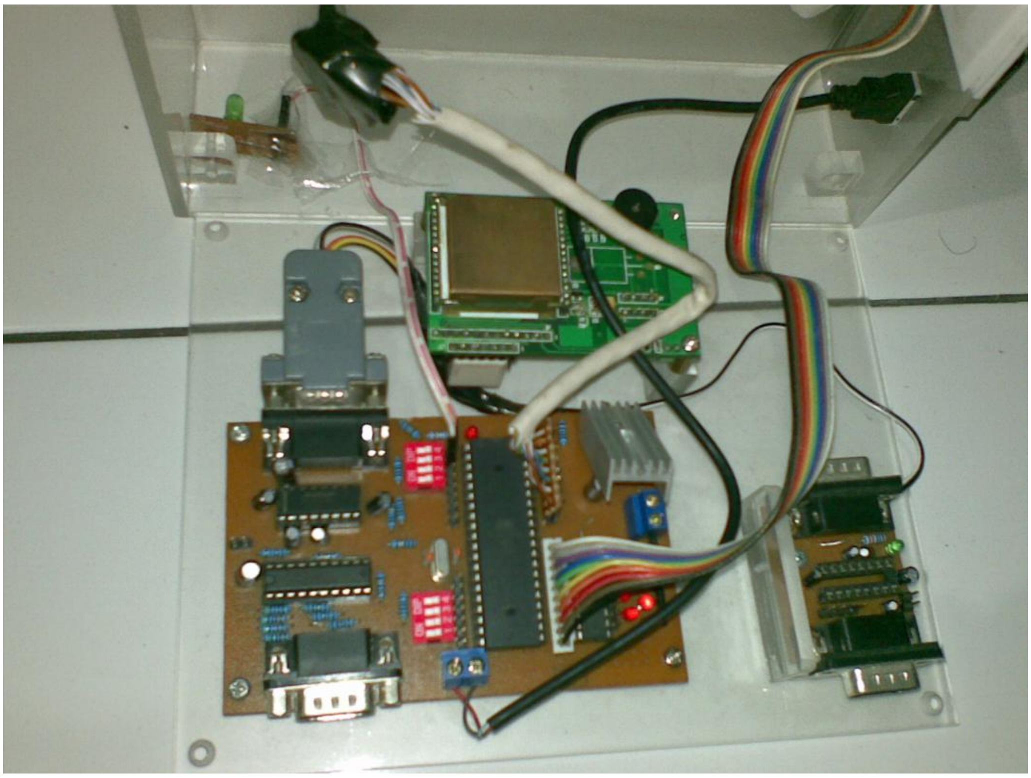
```
            TEMP = getchar();
            return 'I'; //Invalid
      };
   };

if (TEMP == 0x04)
   {
      value = 0;
      val=0;

      val=getchar();
      value = (val << 24);
      val=getchar();
      value = value + (val << 16);
      val=getchar();
      value = value + (val << 8);
      val=getchar();
      value = value + val;
      TEMP = getchar();
      TEMP = getchar();
      return 'S';
    };
            TEMP = getchar();
            TEMP = getchar();
            return 'X'; //Invalid
}



unsigned long SCR_write_value(unsigned char block,long write_val)
{putchar(0x02);
putchar(0x01);
putchar(0x07);
putchar(0x77);
putchar(0x76);
putchar(block);
exoreg=exor(0x07 ,block);
TEMP = (write_val >> 24)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = (write_val >> 16)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = (write_val >> 8)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = write_val & 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
putchar(exoreg);
putchar(0x03);
TEMP = getchar();
TEMP = getchar();
TEMP = getchar();

if (TEMP == 0x01)
   {
   TEMP = getchar();

      if (TEMP == 'F')
      {
            TEMP = getchar();
            TEMP = getchar();
            return 'F'; //Fail
      };

      if (TEMP == 'N')
      {
            TEMP = getchar();
            TEMP = getchar();
            return 'N'; //No tag
        };

        if (TEMP == 'I')
        {
            TEMP = getchar();
            TEMP = getchar();
            return 'I'; //Invalid
        };
        if (TEMP == 'X')
        {
            TEMP = getchar();
            TEMP = getchar();
            return 'X'; //Invalid
        };
    };

if (TEMP == 0x04)
    {
        value = 0;
        val=0;
        val=getchar();
        value = val << 24;
        val=getchar();
        value = value + (val << 16);
        val=getchar();
        value = value + (val << 8);
        val=getchar();
        value = value + val;
        TEMP = getchar();
        TEMP = getchar();
        return value;
     };
            TEMP = getchar();
            TEMP = getchar();
            return 'X'; //Invalid
}

long SCR_dec(char block,long dec_val)
{
putchar(0x02);
putchar(0x01);
putchar(0x06);
putchar(0x2D);
putchar(block);
exoreg=0x2A;
exoreg=exor(exoreg,block);
TEMP = (dec_val >> 24)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = (dec_val >> 16)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = (dec_val >> 8)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = dec_val & 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
putchar(exoreg);
putchar(0x03);

//answer

TEMP = getchar();
TEMP = getchar();
TEMP = getchar();

if (TEMP == 0x01)
    {
    TEMP = getchar();
```

```
        if (TEMP == 'F')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'F'; //Fail
        };

        if (TEMP == 'N')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'N'; //No tag
        };

        if (TEMP == 'I')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'I'; //Invalid
        };
        if (TEMP == 'X')
        {
                TEMP = getchar();
                TEMP = getchar();
                return 'X'; //Invalid
        };
    };

if (TEMP == 0x04)
    {
        value = 0;
        val=0;
        val=getchar();
        value = val << 24;
        val=getchar();
        value = value + (val << 16);
        val=getchar();
        value = value + (val << 8);
        val=getchar();
        value = value + val;
        TEMP = getchar();
        TEMP = getchar();
        return value;
     };
                TEMP = getchar();
                TEMP = getchar();
                return 'X'; //Invalid
}

long SCR_inc(char block,long inc_val)
{
putchar(0x02);
putchar(0x01);
putchar(0x06);
putchar(0x2B);
putchar(block);
exoreg=0x2E;
exoreg=exor(exoreg,block);
TEMP = (inc_val >> 24)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = (inc_val >> 16)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = (inc_val >> 8)& 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
TEMP = inc_val & 0x00FF;
putchar(TEMP);
exoreg=exor(exoreg,TEMP);
```

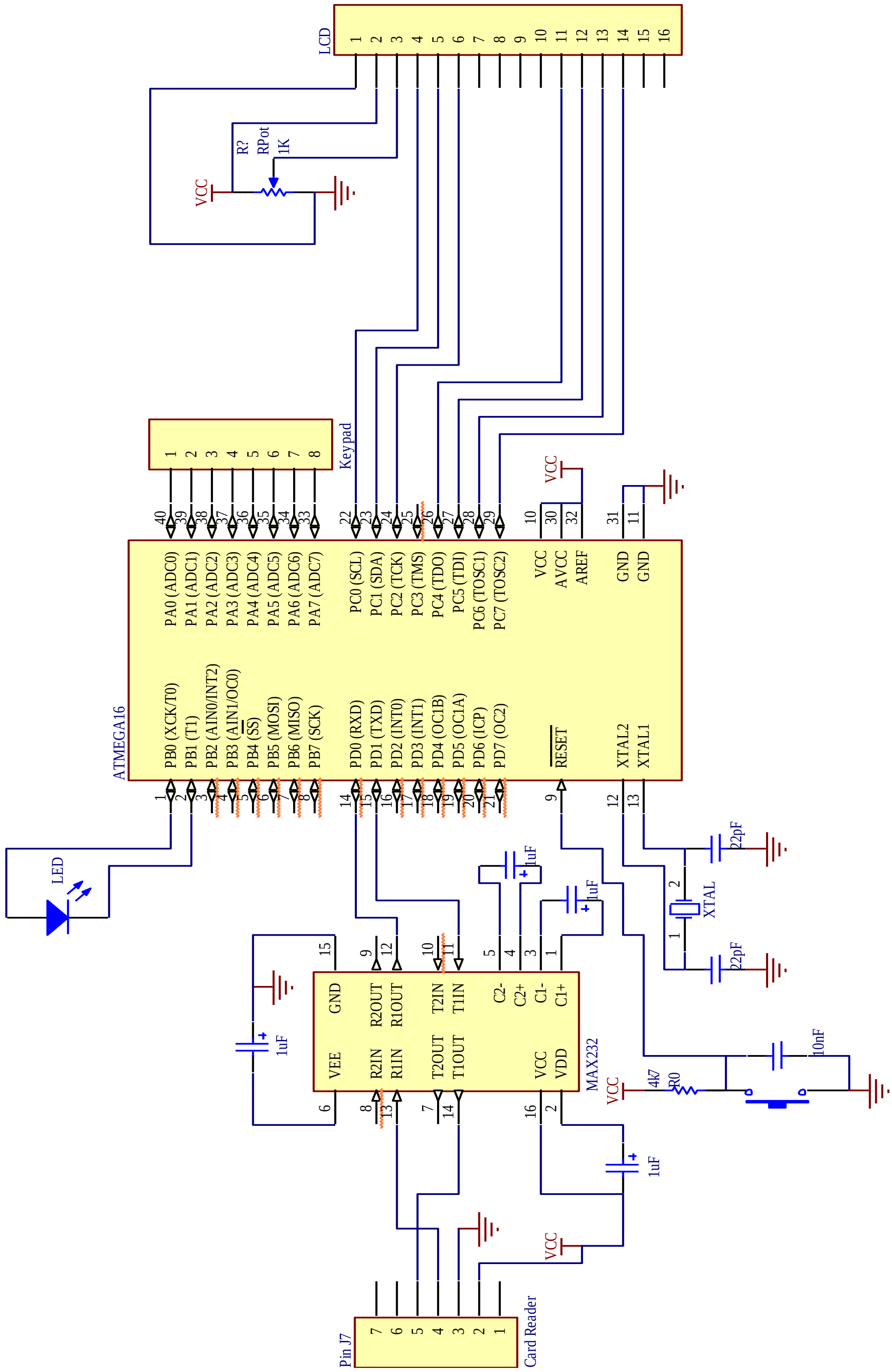
```
putchar(exoreg);
putchar(0x03);

//answer

TEMP = getchar();
TEMP = getchar();
TEMP = getchar();

if (TEMP == 0x01)
  {
  TEMP = getchar();

    if (TEMP == 'F')
    {
        TEMP = getchar();
        TEMP = getchar();
        return 'F'; //Fail
    };

    if (TEMP == 'N')
    {
        TEMP = getchar();
        TEMP = getchar();
        return 'N'; //No tag
    };

    if (TEMP == 'I')
    {
        TEMP = getchar();
        TEMP = getchar();
        return 'I'; //Invalid
    };
    if (TEMP == 'X')
    {
        TEMP = getchar();
        TEMP = getchar();
        return 'X'; //Invalid
    };

  };

if (TEMP == 0x04)
  {
    value = 0;
    val=0;
    val=getchar();
    value = val << 24;
    val=getchar();
    value = value + (val << 16);
    val=getchar();
    value = value + (val << 8);
    val=getchar();
    value = value + val;
    TEMP = getchar();
    TEMP = getchar();
    return value;
   };
        TEMP = getchar();
        TEMP = getchar();
        return 'X'; //Invalid card
}

char scan_keypad(void)
   {
int scankey;
char keypressed =' ';
DDRA = 0x0F;
PORTA = 0xFE;
scankey = PINA&0xf0;
switch (scankey)
```

```
   {
    case 0xE0 : keypressed = '1';
    break;
    case 0xD0 : keypressed = '2';
    break;
    case 0xB0 : keypressed = '3';
    break;
    case 0x70 : keypressed = 'A';
    break;
    }
    PORTA = 0xFD;
    delay_ms(10);
scankey = PINA&0xf0;
switch (scankey)
   {
    case 0xE0 : keypressed = '4';
    break;
    case 0xD0 : keypressed = '5';
    break;
    case 0xB0 : keypressed = '6';
    break;
    case 0x70 : keypressed = 'B';
    break;
    }
    PORTA = 0xFB;
    delay_ms(10);
    scankey = PINA&0xf0;
switch (scankey)
   {
    case 0xE0 : keypressed = '7';
    break;
    case 0xD0 : keypressed = '8';
    break;
    case 0xB0 : keypressed = '9';
    break;
    case 0x70 : keypressed = 'C';
    break;
    }
    PORTA = 0xF7;
    delay_ms(10);
    scankey = PINA&0xf0;
switch (scankey)
   {
    case 0xE0 : keypressed = '*';
    break;
    case 0xD0 : keypressed = '0';
    break;
    case 0xB0 : keypressed = '#';
    break;
    case 0x70 : keypressed = 'D';
    break;
    }
       return keypressed;
        }

  long kov_hextobin(unsigned char hex)
  { switch(hex)
  {case 0x30: bin = 0;
  break;
  case 0x31: bin = 1;
  break;
  case 0x32: bin = 2;
  break;
  case 0x33: bin = 3;
  break;
  case 0x34: bin = 4;
  break;
  case 0x35: bin = 5;
  break;
  case 0x36: bin = 6;
  break;
  case 0x37: bin = 7;
  break;
  case 0x38: bin = 8;
  break;
  case 0x39:bin = 9;
  break;
  }return bin;
  }

  char kov_longtolcd1 (long nilai_a)
  { val1=100000;


  do
  {
  val2=nilai_a/val1;
  nilai_a =nilai_a % val1;
  val1=val1/10;
  }
  while(val2==0);
  lcd_putchar(val2+0x30);
  do
  {
  val2=nilai_a/val1;
  nilai_a =nilai_a % val1;
  val1=val1/10;
  lcd_putchar(val2+0x30);

  }
  while(val1!=0);
  return 0;
  }

  char kov_longtolcd2 (long nilai_b)
  { val1=100000;
  val3=1;
   nilai1[0]=' ';
   nilai1[1]=' ';
   nilai1[2]=' ';
   nilai1[3]=' ';
   nilai1[4]=' ';
   nilai1[5]=' ';
   nilai1[6]=' ';
   nilai1[7]=' ';


  do
  {
  val2=nilai_b/val1;
  nilai_b =nilai_b % val1;
  val1=val1/10;
  }
  while(val2==0);
  nilai1[0]=val2;
  do
  {
  val2=nilai_b/val1;
  nilai_b =nilai_b % val1;
  val1=val1/10;
  nilai1[val3]=val2;
  val3=val3+1;
  }
  while(val1!=0);
  return 0;
  }
```

LAMPIRAN B
FOTO ALAT



LAMPIRAN C
SKEMATIK ALAT


ATMEGA16
PB0 (XCK/T0)
PB1 (T1)
PB2 (AIN0/INT2)
PB3 (AIN1/OC0)
PB4 (SS)
PB5 (MOSI)
PB6 (MISO)
PB7 (SCK)
PD0 (RXD)
PD1 (TXD)
PD2 (INT0)
PD3 (INT1)
PD4 (OC1B)
PD5 (OC1A)
PD6 (ICP)
PD7 (OC2)
RESET
XTAL2
XTAL1
PA0 (ADC0)
PA1 (ADC1)
PA2 (ADC2)
PA3 (ADC3)
PA4 (ADC4)
PA5 (ADC5)
PA6 (ADC6)
PA7 (ADC7)
PC0 (SCL)
PC1 (SDA)
PC2 (TCK)
PC3 (TMS)
PC4 (TDO)
PC5 (TDI)
PC6 (TOSC1)
PC7 (TOSC2)
VCC
AVCC
AREF
GND
GND
40 39 38 37 36 35 34 33
22 23 24 25 26 27 28 29
10 30 32 31 11
1 2 3 4 5 6 7 8
14 15 16 17 18 19 20 21
9 12 13
Keypad
1 2 3 4 5 6 7 8
LCD
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
R?
RPot
1K
VCC
VCC
LED
22pF
22pF
XTAL
2
1
1uF
1uF
1uF
1uF
MAX232
GND
VEE
R2OUT
R1OUT
R2IN
R1IN
T2IN
T1IN
T2OUT
T1OUT
C2-
C2+
C1-
C1+
VCC
VDD
15 9 12 10 11 5 4 3 1
6 8 13 7 14 16 2
VCC
4k7
R0
10nF
VCC
Pin J7
7 6 5 4 3 2 1
Card Reader