

LAMPIRAN

```
/*  
This program was produced by the  
CodewizardAVR V2.03.4 Standard  
Automatic Program Generator  
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com  
*/
```

```
Project :  
Version :  
Date    : 1/27/2016  
Author  :  
Company :  
Comments:
```

```
Chip type      : ATmega16  
Program type   : Application  
Clock frequency : 11.059200 MHz  
Memory model   : Small  
External RAM size : 0  
Data Stack size : 256
```

```
*****/  
while(1)  
{  
  Lcd_gotoxy(0,0);  
  Lcd_putsf(" DIDIK HERMAWAN ");  
}  
  
  Lcd_init(16);  
  lcd_gotoxy(0,0);  
  Lcd_putsf(" Terima Data ");  
  while(1)  
  {  
    getchar()=data;  
    Lcd_gotoxy(0,1);  
    sprintf(buf,"%x h %i d",data,data);  
    Lcd_puts(buf);  
    delay_ms(100);  
  }  
  
  program ADC :  
  #define ADC_VREF_TYPE 0x20  
  // Read the 8 most significant bits  
  // of the AD conversion result  
  unsigned char 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 ADCH;  
  }  
  
  while(1)  
  {  
    dataadc0=read_adc(0);  
    dataadc1=read_adc(1);  
    dataadc2=read_adc(2);  
    dataadc3=read_adc(3);  
    sprintf(buf,"%i%i%i%i",dataadc0, dataadc1, dataadc2,dataadc3);  
    Lcd_gotoxy(0,1);  
    Lcd_puts(buf);  
  }  
}
```

```

while (1)
{
PORTD.2=1;
delay_ms(500);
PORTD.2=0;
delay_ms(500);
}
while (1)
{
PORTD.3=1;
delay_ms(500);
PORTD.3=0;
delay_ms(500);
}

voidscand_data_wireless()
{
char keluar=0;
do
{
dataadc0=read_adc(0);
dataadc1=read_adc(1);
dataadc2=read_adc(2);
dataadc3=read_adc(3);

if(dataadc0==128){data=2;keluar=1;delay_ms(100);}
if(dataadc1==128){data=4;keluar=1;delay_ms(100);}
if(dataadc2==128){data=1;keluar=1;delay_ms(100);}
if(dataadc3==128){data=3;keluar=1;delay_ms(100);}
}
while (keluar ==0);
}

lcd_init(16);
nilai[0] = pointer0;
nilai[1] = pointer1;
nilai[2] = pointer2;
nilai[3] = pointer3;
if ((nilai[0] == 0xff) && (nilai[1] == 0xff) && (nilai[2] == 0xff) &&
(nilai[3] == 0xff))
{
lcd_gotoxy(0,0);
lcd_putsf(" Pass EEPROM ");
lcd_gotoxy(5,1);
delay_ms(300);
j=3;
for (i=0;i<4;i++)
{
do {
scand_data_wireless();delay_ms(100);
} while (data>9);
sprintf(buf,"%x",data);
lcd_puts(buf);
nilai[j-i]=data;
}
pointer0 = nilai[0];
pointer1 = nilai[1];
pointer2 = nilai[2];
pointer3 = nilai[3];
}
else
{
lcd_gotoxy(0,0);
sprintf(buf,"%x%x%x%x%x%x " ,pointer0,pointer1,pointer2,pointer3);
lcd_puts(buf);
delay_ms(1000);
}

lcd_init(16);
nilai[0] = pointer0;
nilai[1] = pointer1;

```

```

nilai[2] = pointer2;
nilai[3] = pointer3;
if ((nilai[0] == 0xff) && (nilai[1] == 0xff) && (nilai[2] == 0xff) &&
(nilai[3] == 0xff))
{
    lcd_gotoxy(0,0);
    lcd_putsf(" Pass EEPROM ");
    lcd_gotoxy(5,1);
    delay_ms(300);
    j=3;
for (i=0;i<4;i++)
{
do {
        scand_data_wireless();delay_ms(100);
        } while (data>9);
sprintf(buf,"%x",data);
    lcd_puts(buf);
    nilai[j-i]=data;
    }
    pointer0 = nilai[0];
    pointer1 = nilai[1];
    pointer2 = nilai[2];
    pointer3 = nilai[3];
}
else
{
    lcd_gotoxy(0,0);
sprintf(buf,"%x%x%x%x%x%x ",pointer0,pointer1,pointer2,pointer3);
    lcd_puts(buf);
    delay_ms(1000);
}

lcd_gotoxy(0,0);
    lcd_putsf(" Password ");
    lcd_gotoxy(5,1);
    delay_ms(100);
    j=3;
for (i=0;i<4;i++)
{
do {
        scand_data_wireless();delay_ms(100);
        } while (data>9);
sprintf(buf,"%x",data);
    lcd_puts(buf);
    nilai[j-i]=data;
    }

if ((nilai[0] == pointer0) && (nilai[1] == pointer1) && (nilai[2] ==
pointer2) && (nilai[3] == pointer3))
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf(" RELAY NC ");
    PORTD.3=1; //RELAY TERHUBUNG
    PORTD.2=0; //BUZZER OFF
    delay_ms(500);
    lcd_clear();
while(a==1)
{
    lcd_gotoxy(0,0);
    lcd_putsf("[A]RELAY NO ");
    lcd_gotoxy(0,1);
    lcd_putsf("[D]New Pass ");
}

lcd_gotoxy(0,0);
    lcd_putsf(" New Password ");
    lcd_gotoxy(5,1);
    delay_ms(300);
    j=3;
for (i=0;i<4;i++)

```

```

do {
    {
        scand_keypad();delay_ms(100);
    } while (data>9);
    sprintf(buf,"%x",data);
    lcd_puts(buf);
    nilai[j-i]=data;
}
pointer0 = nilai[0];
pointer1 = nilai[1];
pointer2 = nilai[2];
pointer3 = nilai[3];
}

else
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf(" Password Error ");
    while(a==1)
    {
        PORTD.1=0; //BUZZER ON
        delay_ms(50);
        PORTD.1=0;
        delay_ms(50);
    }
}

```

