

## DAFTAR PUSTAKA

1. Bedjo, Agus, 2008, "C & AVR", Yogyakarta : Graha Ilmu
2. Anonim, 2009 , "Microcontroler to Sensor Interfacing Techniques", Texas : BiPOM Electronics Inc.
3. Hadi, Mokh. Sholihul, 2008, "Mengenal Mikrokontroler ATMEGA16", Ilmu Komputer.com
4. Pedoman Efisiensi Energi untuk Industri di Asia, diakses tanggal 21 April 2013, <http://energyefficiencyasia.org>
5. LED (light emitting dioda), diakses tanggal 23 April 2013 [http://rasapas.wordpress.com/led\(light emitting dioda\)/Puguh Alakadarnya](http://rasapas.wordpress.com/led(light%20emiting%20dioda)/Puguh%20Alakadarnya)
6. <http://electroniclab.com/Aswan-Hamonangan,Jakarta>, diakses tanggal 20 April 2013
7. Sensor cahaya LDR (Light Dependent Resistor), diakses tanggal 20 April 2013, <http://indomicron.co.cc/elektronika/analog/sensor-cahaya-ldr-light-dependent-resistor>
8. IC ULN2003, diakses tanggal 9 Mei 2013, <http://antomtr.blogspot.com/2010/02/ic-uln-2003.html>
9. Hidayat, Latif, 2011, Skripsi, Universitas Muhammadiyah Yogyakarta
10. Purwanto, Bambang, 2011, Proyek Akhir, Universitas Negeri Yogyakarta
11. Aditya, Rangga, 2012, Skripsi, Universitas Muhammadiyah Yogyakarta
12. <http://www.pln.co.id/dataweb/TTL%202013/Tarif%20Tenaga%20Listrik>

## LIST PROGRAM KENDALI LED OTOMATIS

```
#include <mega16a.h>
#include <stdio.h>
#include <delay.h>
// Alphanumeric LCD functions
#include <alcd.h>
#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;
}
unsigned int mode;
#define LDR read_adc(0)
char lcd_buffer[33];
```

```
// Declare your global variables here
```

```
void main() //fungsi buat memulai LED (memanggil variabel di dalam void main)
```

```
}
```

```
}
```

```
void mode1()
```

```
{
```

```
    lcd_gotoxy(0,1);
```

```
PORTD.6=0;
```

```
}
```

```
if((LDR<210)&&(LDR>=175))
```

```
{
```

```
PORTD.0=1;
```

```
PORTD.1=1;
```

```
PORTD.2=1;
```

```
PORTD.3=1;
```

```
PORTD.4=1;
```

```
PORTD.5=0;
```

```
PORTD.6=0;
```

```
}
```

```
if((LDR<175)&&(LDR>=140))
```

```
{
```

```
PORTD.0=1;
```

```
PORTD.1=1;
```

```
PORTD.2=1;
```

```
PORTD.3=1;
```

```
PORTD.4=0;
```

```
PORTD.5=0;
```

```
PORTD.6=0;
```

```
}
```

```
if((LDR<140)&&(LDR>=105))
```

```
{
```

```
PORTD.0=1;
```

```
PORTD.1=1;
```

```
PORTD.2=1;
```

```
PORTD.3=0;
```

```
PORTD.4=0;
```

```
PORTD.5=0;
```

```
PORTD.6=0;
```

```
}
```

```
if((LDR<105)&&(LDR>=70))
```

```
{
```

```
PORTD.0=1;
```

```
PORTD.1=1;
```

```
PORTD.2=0;
```

```
PORTD.3=0;
```

```
PORTD.4=0;
```

```
PORTD.5=0;
```

```
PORTD.6=0;
```

```
}
```

```
if((LDR<70)&&(LDR>=35))
```

```
{
```

```
PORTD.0=1;
```

```
PORTD.1=0;
```

```
PORTD.2=0;
```

```
PORTD.3=0;
```

```
PORTD.4=0;
```

```
PORTD.5=0;
```

```
    PORTD.6=0;
}
if(LDR<35)
{
    PORTD.0=0;
    PORTD.1=0;
    PORTD.2=0;
    PORTD.3=0;
    PORTD.4=0;
    PORTD.5=0;
    PORTD.6=0;
}
}
unsigned int timer;
void mode2()
{
    while(mode==2)
    {
        lcd_gotoxy(0,1);
        sprintf(lcd_buffer,"mode=%d t=%d ",mode,timer);
        lcd_puts(lcd_buffer);
        PORTD.7=~PORTD.7;
        timer++;
        if(timer<240)
        {
```