

# DT-AVR

## DT-AVR *Application Note*

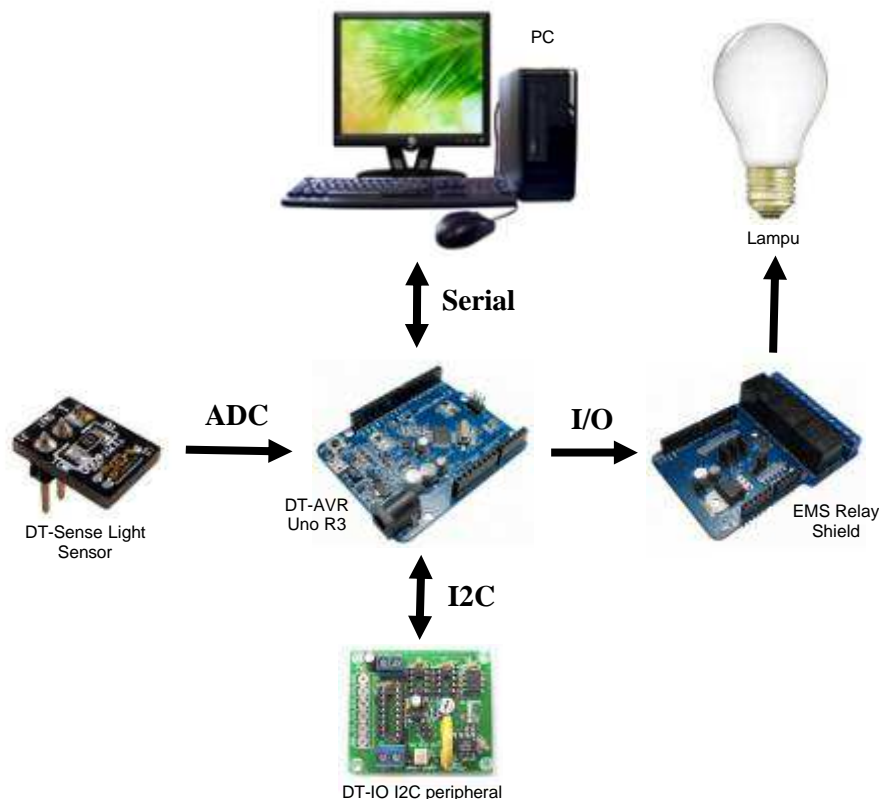
### AN230 – Lampu Taman Otomatis

Oleh: Tim IE

Aplikasi relay secara luas telah banyak digunakan terutama pada bidang robotika sebagai *driver* motor DC. Selain itu, relay juga bisa diaplikasikan untuk mengontrol peralatan listrik berarus AC, salah satunya lampu. Pada aplikasi kali ini akan digunakan lampu yang dikontrol penyalanya secara otomatis berdasarkan parameter sensor cahaya dan pewaktuan RTC (*Real Time Clock*). Sehingga diharapkan lampu secara otomatis akan menyala ketika malam hari dan akan padam di siang harinya. Aplikasi ini sangat cocok diterapkan untuk lampu-lampu yang penempatannya di luar ruangan, lampu taman misalnya. Dalam pengembangannya, aplikasi ini memerlukan beberapa perangkat sebagai berikut:

- 1x DT-AVR Uno R3
- 1x EMS Relay Shield
- 1x DT-Sense Light Sensor
- 1x DT-IO I<sup>2</sup>C Peripheral
- 1x Lampu
- Kabel secukupnya

Adapun blok diagram dari aplikasi ini adalah sebagai berikut :



**Gambar 1**  
**Blok Diagram AN230**

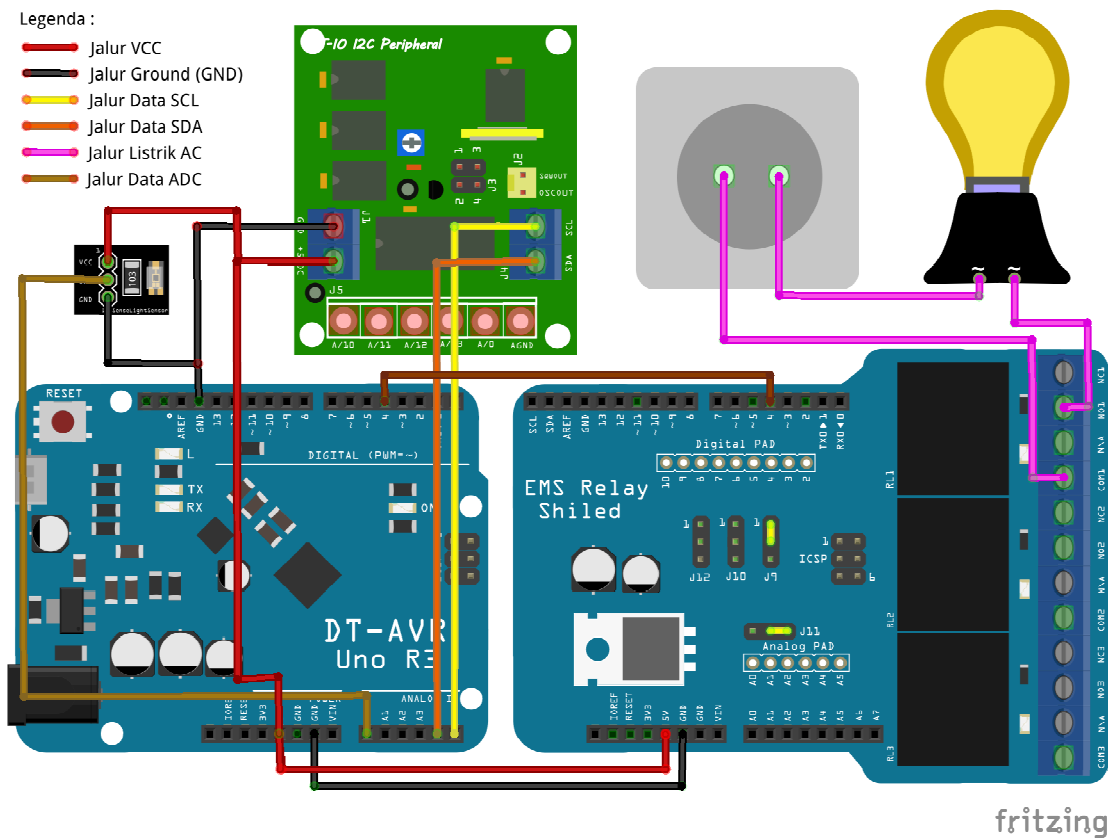
Hubungan antara DT-AVR Uno R3 dengan EMS Relay Shield, DT-IO I<sup>2</sup>C Peripheral, dan DT-Sense Light Sensor adalah sebagai berikut:

| DT-AVR Uno R3 | EMS Relay Shield | DT-IO I <sup>2</sup> C Peripheral | DT-Sense Light Sensor |
|---------------|------------------|-----------------------------------|-----------------------|
| GND (J4)      | GND (J4)         | PIN GND (J3)                      | PIN 3 - GND (J1)      |
| VCC (J4)      | VCC (J4)         | PIN VCC (J3)                      | PIN 1 - VCC (J1)      |
| PIN 4 (J2)    | PIN 4 (J3)       | -                                 | -                     |
| PIN A0 (J5)   | -                | -                                 | PIN 2 - OUT (J1)      |
| PIN A4 (J5)   | -                | PIN SCK (J2)                      | -                     |
| PIN A5 (J5)   | -                | PIN SCL (J2)                      | -                     |

\* pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah konfigurasi pin pada program

**Tabel 1**  
**Hubungan DT-AVR Uno R3 dengan EMS Relay Shield, DT-IO I<sup>2</sup>C Peripheral, dan DT-Sense Light Sensor**

Hal pertama yang perlu dilakukan adalah melakukan koneksi antara modul EMS Relay Shield, modul DT-IO I<sup>2</sup>C Peripheral, dan modul DT-Sense Light Sensor dengan DT-AVR Uno R3 seperti pada **Tabel 1**. Ilustrasi koneksi antar modul tersebut adalah seperti pada **Gambar 2**.



\* Ilustrasi gambar dibuat menggunakan aplikasi [Fritzing](#)

**Gambar 2**  
**Ilustrasi Koneksi antara Modul EMS Relay Shield, DT-IO I<sup>2</sup>C Peripheral, dan DT-Sense Light Sensor dengan DT-AVR Uno R3**

Jika semua koneksi telah terhubung, maka coba pastikan sekali lagi bahwa pin-pin tersebut telah terhubung dengan benar. Terutama untuk pin VCC dan GND, jangan sampai polaritasnya terbalik. Karena hal tersebut dapat merusak komponen terutama IC. Dan perlu perhatikan pula bahwa pada modul EMS Relay Shield terdapat *jumper*


J11 yang berguna untuk memilih sumber daya yang akan dipakai untuk modul dan juga *jumper* J9 untuk mengatur pin input kontrol relay (keterangan lebih lanjut dapat dilihat pada panduan manual EMS Relay Shield atau **Gambar 2**). Pada aplikasi ini, *jumper* J11 diatur agar modul menggunakan sumber tegangan +5V yang berasal dari regulator DT-AVR UNO R3 dan *jumper* J9 diatur agar pin kontrol relay RL1 dari modul EMS Relay Shield terhubung dengan pin digital I/O pin 4 dari DT-AVR Uno R3.

Kemudian satu hal penting lagi yang perlu diperhatikan yaitu aplikasi ini memerlukan sumber tegangan AC 220 volt sebagai sumber daya untuk lampu. Sumber tersebut diambil dari stop kontak, yang tentunya akan berbahaya jika salah dalam merangkai, sebab bisa menyebabkan arus pendek/konslet. Oleh karenanya, untuk menghindari resiko tersebut maka sangat dianjurkan untuk memeriksa ulang keseluruhan rangkaian sebelum *test drive*. Pastikan semua rangkaian tersebut telah terpasang dengan benar seperti yang terlihat pada **Gambar 2** dan **Tabel 1**.

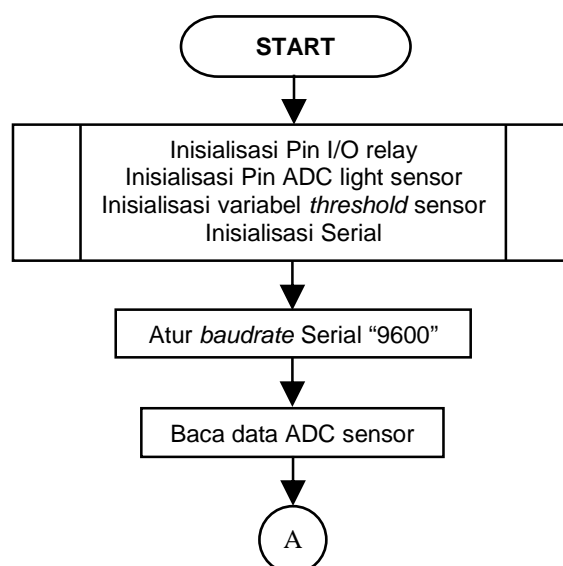
Aplikasi ini memakai dua buah sumber referensi sebagai patokan/tanda untuk mengaktifkan dan menonaktifkan relay, yaitu sensor cahaya dengan menggunakan DT-Sense Light Sensor dan pewaktuan RTC dengan DT-IO I<sup>2</sup>C Peripheral. Sebetulnya dengan hanya menggunakan salah satu dari kedua referensi tersebut sudah bisa, namun disini akan dibandingkan efektifitas hasilnya dari kedua sumber referensi tersebut. Oleh karenanya, pada aplikasi ini akan dicoba tiga kali percobaan dimana dua percobaan dengan masing-masing sumber referensi tersebut dan percobaan yang ketiga dengan kombinasi keduanya.

## **Percobaan Pertama**

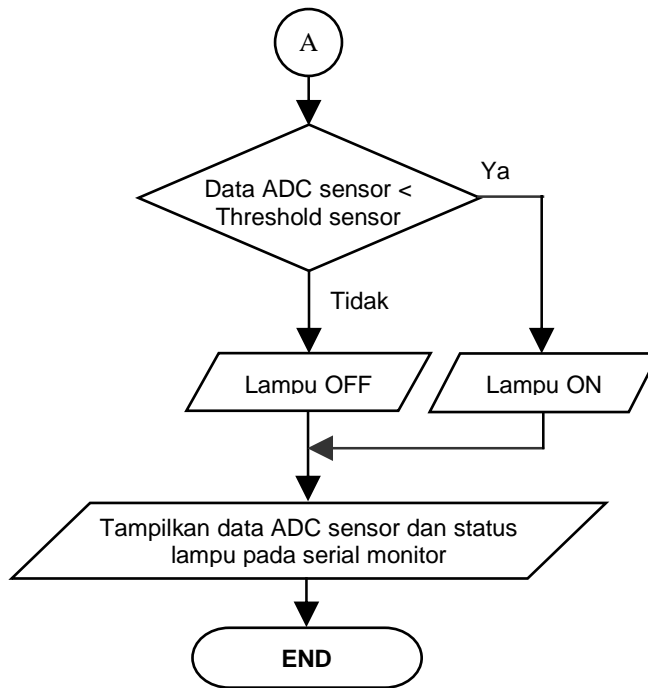
Pada percobaan pertama ini akan digunakan sensor cahaya sebagai referensi untuk mengaktifkan dan menonaktifkan relay, yang mana logikanya jika sensor mendapati kondisi gelap (misal malam hari), maka relay akan diaktifkan sehingga lampu akan menyala. Sebaliknya, jika sensor mendapati kondisi terang (misal siang hari), maka relay akan dinonaktifkan sehingga lampu akan padam. Sensor cahaya yang dipakai adalah modul DT-Sense Light Sensor. Untuk melakukan percobaan ini, ikuti langkah-langkahnya sebagai berikut:

1. Hubungkan DT-AVR Uno R3 dengan PC/Laptop menggunakan kabel USB.
2. Jalankan program Arduino IDE.
3. Atur *board* yang akan digunakan dengan memilih **Tools – Board – Arduino Uno**.
4. Atur *port* serial yang akan digunakan pada **Tools – Port – COMx** (COMx disesuaikan dengan *port serial* DT-AVR Uno R3 yang terlihat pada *device manager*).
5. Buka *file* program **AN230\_Test1.ino**. Lalu lakukan perintah *upload sketch* dengan menekan tombol  (Jika terjadi kendala saat proses *upload*, coba lakukan *troubleshooting* dengan bantuan yang merujuk pada laman <https://www.arduino.cc/en/guide/troubleshooting>).
6. Selanjutnya buka *Terminal/Serial Monitor* dan atur *baudrate* pada nilai **9600**.
7. Amati perubahan nyala lampu ketika cahaya sekeliling berubah dari terang ke redup dan sebaliknya.

**F**lowchart sistem dari program “AN230\_Test1.ino” adalah sebagai berikut.



\* Flowchart berlanjut pada halaman berikutnya



**Gambar 3**  
**Flowchart Program AN230\_Test1.ino**

**P**enjelasan singkat mengenai *flowchart* tersebut adalah sebagai berikut:

1. Inisialisasi pin I/O relay, pin ADC, dan variabel *threshold* sensor.

```

int Relay1 = 4; // Pin to control the Relay RL1 of "EMS Relay Shield"
int pin_lightSensor = A0; // pin to get ADC data of "DT-Sense Light Sensor"
int threshold_lightSensor = 3; // variabel acuan sensor cahaya
  
```

2. Inisialisasi *register output* dari pin I/O relay.

```

pinMode(Relay1, OUTPUT); // pin Relay1 to be output
digitalWrite(Relay1, LOW); // Relay RL1 off, so the lamp should be OFF
  
```

3. Inisialisasi Serial dengan *baudrate* 9600.

```

Serial.begin(9600);
  
```

4. Baca data ADC sensor dari modul DT-Sense Light Sensor.

```

int value_lightSensor = analogRead(pin_lightSensor);
  
```

5. Bandingkan data ADC sensor dengan nilai *threshold*, jika nilai ADC lebih kecil maka relay akan diaktifkan sehingga lampu menyala. Kondisi ini akan terjadi ketika sensor tidak mendapatkan cukup cahaya (gelap).

```

if(value_lightSensor < threshold_lightSensor) { // pada kondisi gelap
    digitalWrite(Relay1, HIGH); // Relay1 aktif sehingga Lampu nyala
    Serial.println("Nyala");
}
else { // pada kondisi terang
    digitalWrite(Relay1, LOW); // Relay1 nonaktif sehingga Lampu padam
    Serial.println("Padam");
}
  
```

6. Menampilkan data sensor dan status lampu pada serial monitor.


```

Serial.print("Nilai light Sensor : ");
Serial.println(value_lightSensor);
Serial.print("Status Lampu : ");
  
```

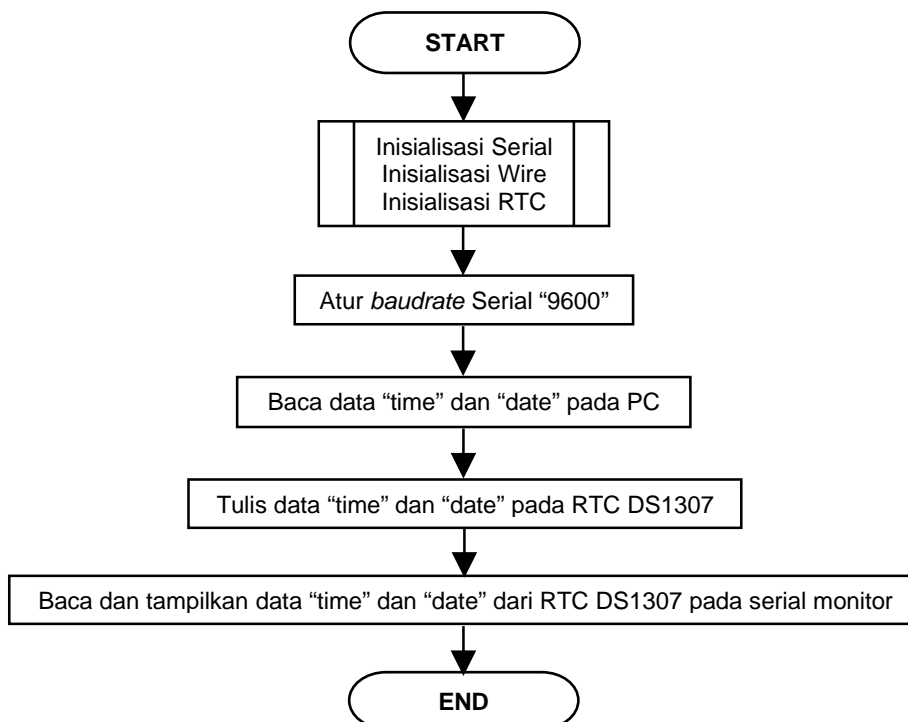
**Catatan** : Satu hal yang perlu diketahui yaitu bahwa aplikasi percobaan pertama ini memiliki kelemahan. Kelemahannya adalah ketika cuaca sedang mendung gelap atau hujan, otomatis jumlah intensitas cahaya yang tertangkap oleh sensor akan sangat sedikit sehingga nilainya mungkin akan berada di bawah *threshold*. Hal ini dapat menyebabkan relay aktif sehingga lampu menyala (padahal masih siang hari). Untuk menghindari kondisi tersebut, nilai variabel *threshold* pada kode program harus dinaikkan sampai pada nilai tertentu dimana sensor telah bisa membedakan apakah saat itu sedang cuaca mendung gelap di siang hari atautkah memang sudah malam hari. Variabel *threshold* yang dimaksud adalah variabel *threshold\_lightSensor*.

## **P**ercobaan Kedua

Pada percobaan kedua, yang mempengaruhi secara langsung aktif tidaknya relay adalah pewaktuan *Real Time Clock* (RTC), yang mana sederhananya kita membuat jadwal *timer* yang mengondisikan kapan relay akan aktif dan kapan relay akan nonaktif. Logikannya adalah jika waktu RTC telah mencapai waktu “timer ON” maka relay akan aktif sehingga lampu akan menyala. Dan sebaliknya ketika waktu RTC telah mencapai waktu “timer OFF” maka relay akan nonaktif dan lampu akan padam. Untuk melakukan percobaan ini, pertama-tama perlu mengatur pewaktuan RTC yang terdapat pada modul **DT-IO I<sup>2</sup>C Peripheral** agar sesuai dengan waktu saat ini. Berikut langkah-langkahnya:

1. Atur jam pada PC/laptop dengan benar.
2. Hubungkan DT-AVR Uno R3 dengan PC/Laptop menggunakan kabel USB.
3. Tambahkan *library RTCLib* pada *folder library /My documents/Arduino/libraries/*
4. Jalankan program Arduino IDE.
5. Atur *board* yang akan digunakan dengan memilih **Tools – Board – Arduino Uno**.
6. Atur *port* serial yang akan digunakan pada **Tools – Port – COMx**.
7. Buka *file* program **AN230\_Setting\_Time.ino**. Lalu lakukan perintah *upload sketch* dengan menekan tombol .
8. Selanjutnya buka *Serial Monitor* dan atur *baudrate* dengan nilai **9600**.
9. Lihat jam yang terdapat pada komputer kemudian bandingkan kesesuaiannya dengan pewaktuan RTC yang terlihat pada serial monitor.

**F**lowchart dari sistem pada program “AN230\_Setting\_Time.ino” adalah sebagai berikut.



\* IC DS1307 terpasang pada modul DT-IO I<sup>2</sup>C Peripheral

**Gambar 4**  
**Flowchart Program AN230\_Setting\_Time.ino**

Flowchart program tersebut bertujuan untuk melakukan konfigurasi pewaktuan RTC pada IC DS1307 dengan format YYYY/MM/DD HH:MM:SS. Pewaktuan RTC tersebut diperlukan sebagai salah satu parameter referensi untuk program selanjutnya.

**P**enjelasan singkat mengenai *flowchart* tersebut adalah sebagai berikut:

1. Deklarasi RTC.

```
RTC_DS1307 rtc;
```

2. Inisialisasi wire, RTC, dan serial dengan *baudrate* 9600.

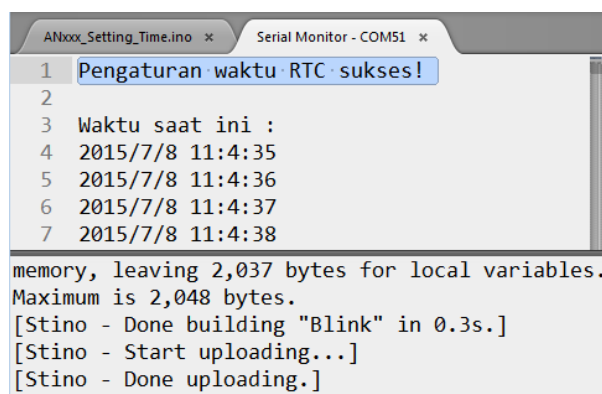
```
Serial.begin(9600);  
Wire.begin();  
rtc.begin();
```

3. Ketika kode program di-*compile*, pengaturan jam dan tanggal yang terdapat pada PC akan dibaca kemudian akan ditulis pada IC RTC DS1307 yang terpasang pada modul DT-IO I<sup>2</sup>C Peripheral sesaat setelah program di-*upload* pada board DT-AVR Uno R3.

```
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
```

4. Selanjutnya data jam dan tanggal yang terdapat pada IC RTC DS1307 dibaca oleh mikrokontroler dan kemudian ditampilkan ke PC melalui serial monitor.

```
DateTime now = rtc.now();  
// menampilkan waktu dgn format YYYY/MM/DD HH:MM:SS  
Serial.print(now.year(), DEC);  
Serial.print('/');  
Serial.print(now.month(), DEC);  
Serial.print('/');  
Serial.print(now.day(), DEC);  
Serial.print(' ');  
Serial.print(now.hour(), DEC);  
Serial.print(':');  
Serial.print(now.minute(), DEC);  
Serial.print(':');  
Serial.print(now.second(), DEC);  
Serial.println();
```




```
ANxxx_Setting_Time.ino x Serial Monitor - COM51 x  
1 Pengaturan waktu RTC sukses!  
2  
3 Waktu saat ini :  
4 2015/7/8 11:4:35  
5 2015/7/8 11:4:36  
6 2015/7/8 11:4:37  
7 2015/7/8 11:4:38  
  
memory, leaving 2,037 bytes for local variables.  
Maximum is 2,048 bytes.  
[Stino - Done building "Blink" in 0.3s.]  
[Stino - Start uploading...]  
[Stino - Done uploading.]
```

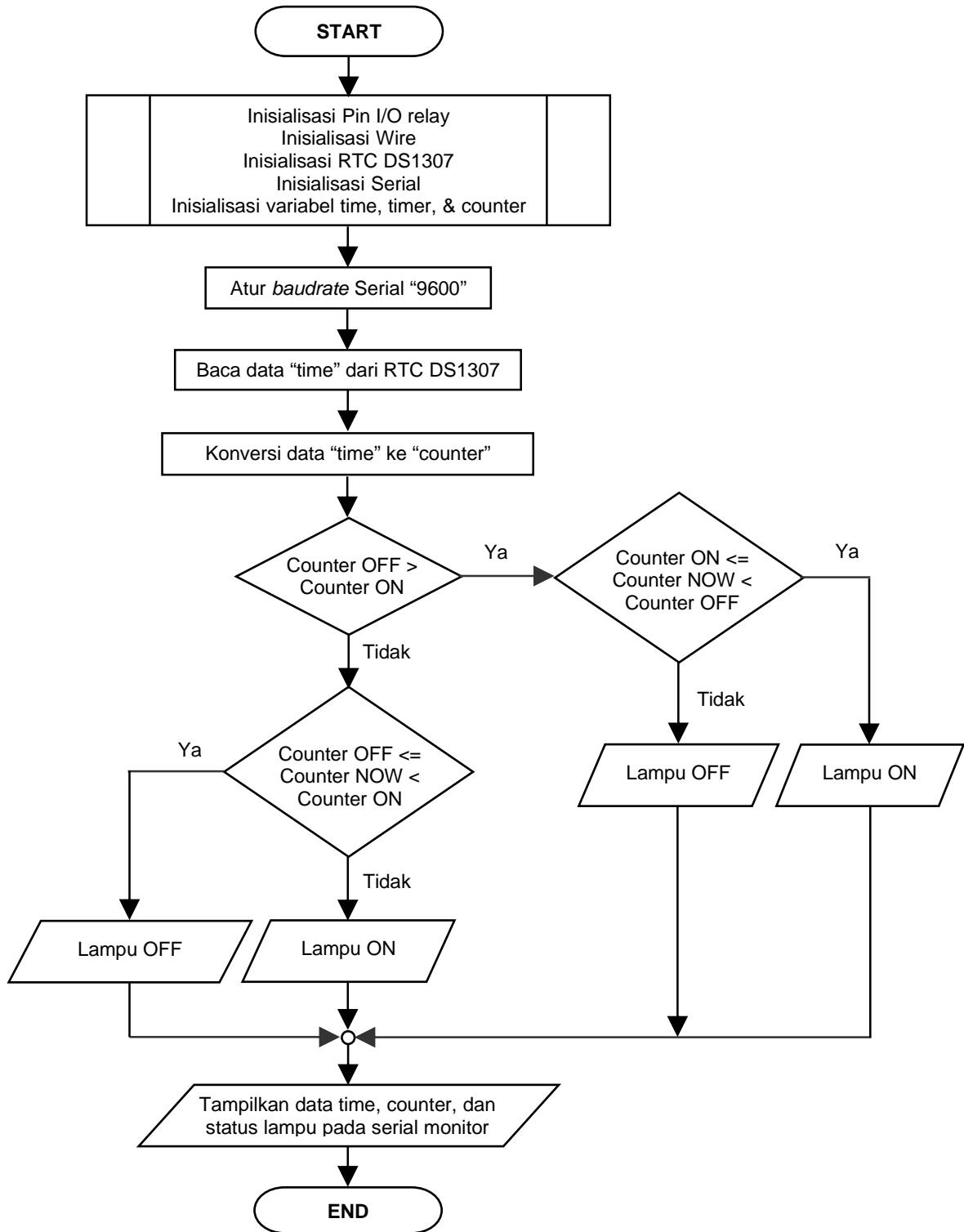
**Gambar 5**  
Hasil Pengaturan Pewaktuan RTC yang Tertampil pada Serial Monitor

Jika tahap ini sukses maka sekarang saatnya melakukan percobaan kedua, yaitu membuat aplikasi kontrol relay berdasarkan pewaktuan RTC. Langkah-langkahnya adalah sebagai berikut:

1. Pastikan catu daya atau sumber listrik yang berasal dari stop kontak telah terpasang dengan benar pada pin yang telah ditentukan pada modul EMS Relay Shield (lihat **Gambar 2**).
2. Hubungkan DT-AVR Uno R3 pada PC/Laptop menggunakan kabel USB.
3. Jalankan program Arduino IDE.

4. Atur *board* yang akan digunakan dengan memilih **Tools – Board – Arduino Uno**.
5. Atur *port* serial yang akan digunakan pada **Tools – Port – COMx**.
6. Buka *file* program **AN230\_Test2.ino**. Lalu lakukan perintah *upload sketch* dengan menekan tombol .
7. Selanjutnya buka *Terminal/Serial Monitor* dan atur *baudrate* pada nilai **9600**.
8. Amati nyala lampu ketika sore hari pukul 17:05:00 dan pada pagi hari pukul 05:30:00.

**F**lowchart sistem dari program “AN230\_Test2.ino” adalah sebagai berikut.



\* IC DS1307 terpasang pada modul DT-IO I<sup>2</sup>C Peripheral

**Gambar 6**  
**Flowchart Program AN230\_Test2.ino**

**P**enjelasan singkat mengenai *flowchart* tersebut adalah sebagai berikut:

1. Deklarasi variabel RTC.

```
RTC_DS1307 rtc;
```

2. Inisialisasi pin I/O relay dan *timer*. *Timer* yang dimaksud adalah *timer ON* dan *timer OFF*. *Timer ON* merupakan waktu yang menunjukkan kapan *relay* akan aktif dan *timer OFF* merupakan waktu yang menunjukkan kapan *relay* akan nonaktif. *Timer ON* diwakili oleh variabel *on\_hour*, *on\_minute*, *on\_second*. Sedangkan *timer OFF* diwakili oleh variabel *off\_hour*, *off\_minute*, *off\_second*.

```
int Relay1 = 4; //DT-AVR Uno Pin to control the Relay RL1 of "EMS Relay Shield"
```

```
// Konfigurasi timer ON pada jam 17:05:00 (sore hari)
```

```
int on_hour = 17;
```

```
int on_minute = 5;
```

```
int on_second = 0;
```

```
// Konfigurasi timer OFF pada jam 05:30:00 (pagi hari)
```

```
int off_hour = 5;
```

```
int off_minute = 30;
```

```
int off_second = 0;
```

3. Inisialisasi *register output* dari pin I/O relay.

```
pinMode(Relay1, OUTPUT); // pin Relay1 to be output
```

```
digitalWrite(Relay1, LOW); // Relay RL1 off, so the lamp should be OFF
```

4. Inisialisasi wire, RTC, dan serial dengan *baudrate* 9600.

```
Wire.begin();
```

```
rtc.begin();
```

```
Serial.begin(9600);
```

5. Baca data "time" dari register IC RTC DS1307. Data "time" yang terbaca merupakan waktu *real time* saat itu, dan pada kode program disebut dengan *time NOW* yang diwakili oleh variabel *now\_hour*, *now\_minute*, *now\_second*.

```
DateTime now = rtc.now();
```

```
int now_hour = now.hour();
```

```
int now_minute = now.minute();
```

```
int now_second = now.second();
```

6. Lalu konversikan data *time* baik itu *time NOW*, *timer ON*, dan *timer OFF* menjadi data *counter* untuk memudahkan pemahaman dalam pemrograman. Perhitungan *counter* tersebut pada dasarnya mengubah semua satuan waktu (jam, menit, detik) menjadi detik. Sederhananya, dalam 1 jam ada 60 menit 60 detik (60\*60 detik) dan dalam 1 menit ada 60 detik. Itu artinya jika pada saat itu jam menunjukkan pukul 00:00:00 maka jika dikonversi menjadi detik nilai *counter*-nya adalah 0 detik. Dan jika saat itu waktu menunjukkan pukul 23:59:59 maka nilai *counter*-nya adalah 86.399 detik.

```
long counter_NOW = (long)now_hour*60*60 + now_minute*60 + now_second;
```

```
long counter_ON = (long)on_hour*60*60 + on_minute*60 + on_second;
```

```
long counter_OFF = (long)off_hour*60*60 + off_minute*60 + off_second;
```

7. Kemudian berdasarkan data *counter* tersebut, akan dapat ditentukan dengan mudah kapan relay harus aktif dan kapan harus dinonaktifkan. Proses penentuan tersebut terdapat pada cuplikan program berikut.

```
if(counter_OFF > counter_ON) {
```

```
    // terjadi jika nilai counter saat timer OFF > nilai counter saat timer ON,
```

```
    // misal ON pada pukul 17:15 sore & OFF pada pukul 21:45 malam hari itu juga
```

```
    if(counter_NOW >= counter_ON && counter_NOW < counter_OFF) {
```

```
        digitalWrite(Relay1, HIGH); // Relay1 aktif -> Lampu nyala
```

```
        Serial.println("Nyala");
```

```
    }
```

```
    else {
```

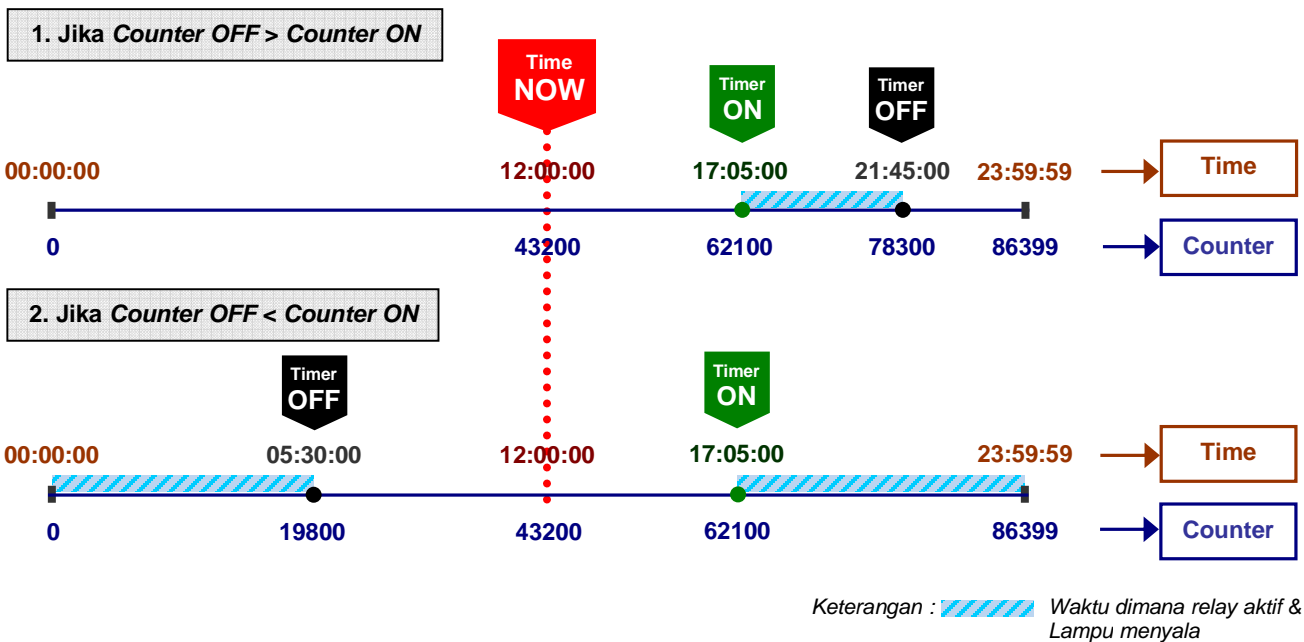


```

digitalWrite(Relay1, LOW); // Relay1 nonaktif -> Lampu padam
Serial.println("Padam");
}
}
else { // logikanya sama dengan if(counter_ON > counter_OFF)
// terjadi jika nilai counter saat timer ON > nilai counter saat timer OFF,
// misal ON pada pukul 17:05 sore & OFF pada pukul 05:30 keesokan paginya
if(counter_NOW >= counter_OFF && counter_NOW < counter_ON) {
digitalWrite(Relay1, LOW); // Relay1 nonaktif -> Lampu padam
Serial.println("Padam");
}
else {
digitalWrite(Relay1, HIGH); // Relay1 aktif -> Lampu nyala
Serial.println("Nyala");
}
}
}

```

Berdasarkan program di atas, terdapat dua kondisi yang berbeda untuk menentukan aktif tidaknya sebuah relay yakni berdasarkan nilai dari variabel *counter\_ON* dan *counter\_OFF*. Untuk lebih jelasnya dapat dilihat pada ilustrasi *timeline* pada **Gambar 7**.




**Gambar 7**  
**Ilustrasi Timeline Hubungan antara Counter NOW, Counter ON, dan Counter OFF**

- Menampilkan data *time*, *counter*, dan status lampu pada serial monitor. Hal ini akan sangat memudahkan kita dalam proses *monitoring* dan *troubleshooting* jika terjadi masalah.

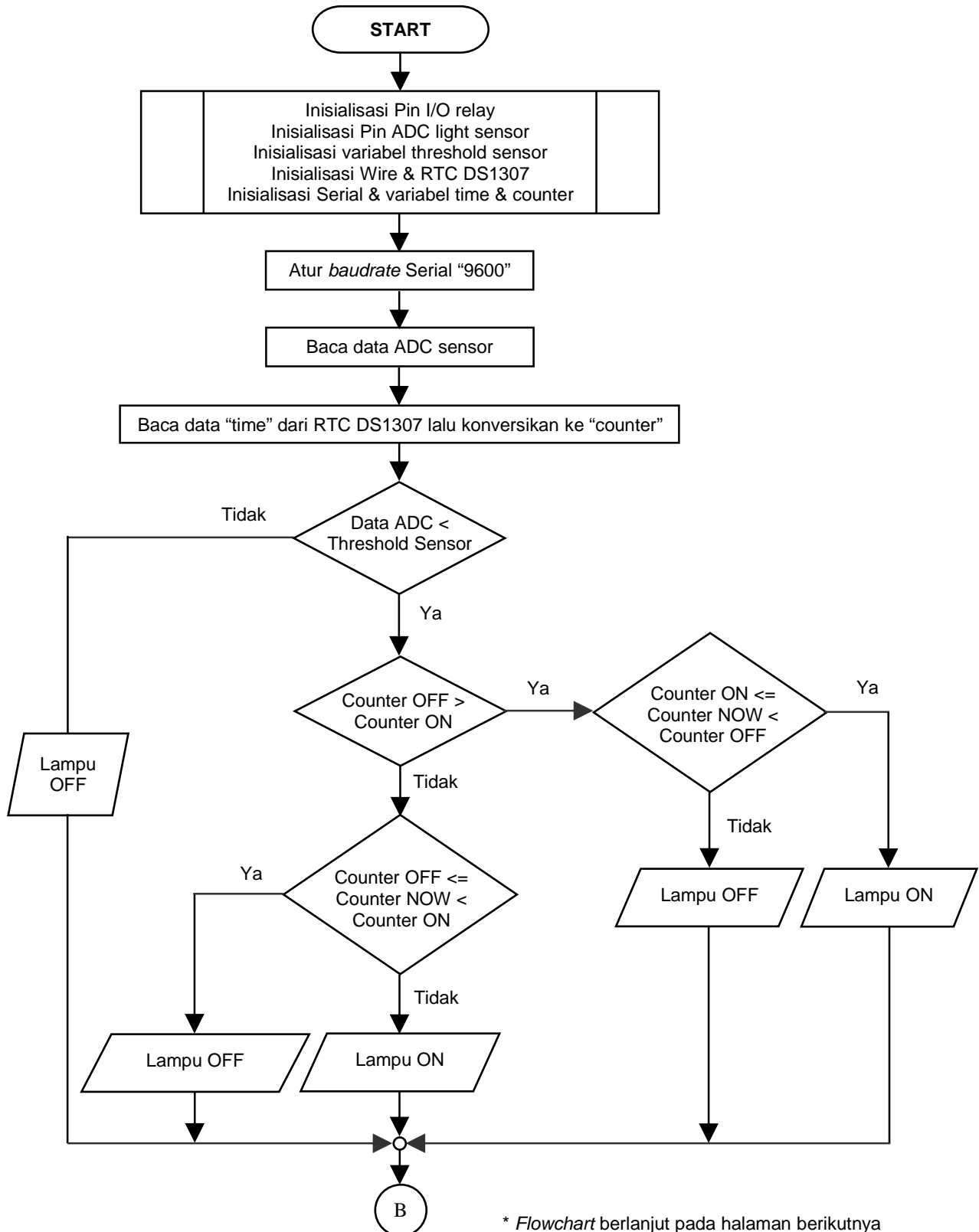
**Catatan :** Di Indonesia, terdapat perbedaan antara waktu terbit dan waktu terbenamnya matahari pada bulan-bulan tertentu. Jadi kemungkinan pada bulan-bulan tersebut akan terjadi kondisi dimana relay akan aktif ketika sore hari masih agak terang. Untuk mengatasi hal tersebut kita bisa menginisialisasikan nilai *timer ON* dan *timer OFF* yang berbeda-beda di tiap bulan-bulan tersebut.

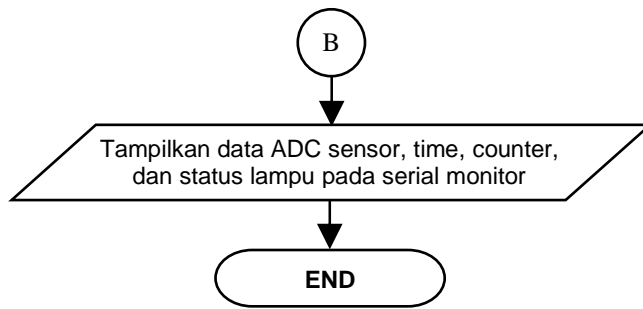
## **P**ercobaan Ketiga

Percobaan ini merupakan gabungan dari kedua percobaan sebelumnya. Dengan demikian, yang mempengaruhi aktif tidaknya relay adalah sensor cahaya dan juga pewaktuan RTC. Masing-masing memiliki peran tersendiri. Sensor cahaya berperan dalam menentukan apakah kondisi sudah gelap atau bukan, sedangkan RTC membantu sensor cahaya memastikan apakah pada saat itu memang sudah menjelang malam atau mungkin hanya mendung gelap di siang bolong. Untuk memulai percobaan ini, berikut adalah langkah-langkahnya:

1. Hubungkan DT-AVR Uno R3 pada PC/Laptop menggunakan kabel USB.
2. Jalankan program Arduino IDE.
3. Buka *file* program **AN230\_Test3.ino**. Lalu lakukan perintah *upload sketch* dengan menekan tombol .
4. Selanjutnya buka *Terminal/Serial Monitor* dan atur *baudrate* pada nilai **9600**.
5. Amati nyala lampu ketika sore hari pukul 17:05:00 dan pada pagi hari pukul 05:30:00.

**F**lowchart sistem dari program “AN230\_Test3.ino” adalah sebagai berikut.





\* IC RTC DS1307 terpasang pada modul DT-IO I<sup>2</sup>C Peripheral

**Gambar 8**  
**Flowchart Program AN230\_Test3.ino**

**P**enjelasan singkat mengenai *flowchart* pada **Gambar 8** adalah sebagai berikut:

1. Deklarasi variabel RTC.

```
RTC_DS1307 rtc;
```

2. Inisialisasi pin I/O relay, pin ADC, dan variabel threshold sensor. Perlu digarisbawahi, nilai variabel threshold ini mungkin akan berbeda, tergantung peletakan sensor dan seberapa banyak cahaya yang mampu ditangkap.

```
int Relay1 = 4; //DT-AVR Uno Pin to control the Relay RL1 of "EMS Relay Shield"
int pin_lightSensor = A0; // pin to get ADC data of "DT-Sense Light Sensor"
int threshold_lightSensor = 3; // variabel pembatas sensor cahaya
```

3. Inisialisasi variabel *timer*. *Timer* yang dimaksud adalah *timer ON* dan *timer OFF*. *Timer ON* merupakan waktu yang menunjukkan kapan *relay* akan aktif dan *timer OFF* merupakan waktu yang menunjukkan kapan *relay* akan nonaktif. *Timer ON* diwakili oleh variabel *on\_hour*, *on\_minute*, *on\_second*. Sedangkan *timer OFF* diwakili oleh variabel *off\_hour*, *off\_minute*, *off\_second*.

```
// Konfigurasi timer ON pada jam 17:05:00 (sore hari)
int on_hour = 17;
int on_minute = 5;
int on_second = 0;

// Konfigurasi timer OFF pada jam 05:30:00 (pagi hari)
int off_hour = 5;
int off_minute = 30;
int off_second = 0;
```

4. Inisialisasi *register output* dari pin I/O relay.

```
pinMode(Relay1, OUTPUT); // pin Relay1 to be output
digitalWrite(Relay1, LOW); // Relay RL1 off, so the lamp should be OFF
```

5. Inisialisasi *wire*, *RTC*, dan *serial* dengan *baudrate* 9600.

```
Wire.begin();
rtc.begin();
Serial.begin(9600);
```

6. Baca data "time" dari register IC RTC DS1307. Data "time" yang terbaca merupakan waktu *real time* saat ini, dan pada aplikasi ini disebut sebagai *time NOW* dan diwakili oleh variabel *now\_hour*, *now\_minute*, *now\_second*.

```
DateTime now = rtc.now();
int now_hour = now.hour();
int now_minute = now.minute();
int now_second = now.second();
```

7. Lalu konversikan semua data, baik itu *time NOW*, *timer ON*, dan *timer OFF*, menjadi data *counter* untuk memudahkan pemahaman dalam pemrograman. Perhitungan *counter* tersebut pada dasarnya mengubah semua satuan waktu (jam, menit, detik) menjadi detik. Sederhananya, dalam 1 jam ada 60 menit 60 detik (60\*60 detik) dan dalam 1 menit ada 60 detik. Itu artinya jika pada saat itu waktu menunjukkan pukul 00:00:00 maka jika dikonversi menjadi detik nilai *counter*-nya adalah 0 detik. Dan jika saat itu waktu menunjukkan pukul 23:59:59 maka nilai *counter*-nya adalah 86.399 detik.

```
long counter_NOW = (long)now_hour*60*60 + now_minute*60 + now_second;
long counter_ON = (long)on_hour*60*60 + on_minute*60 + on_second;
long counter_OFF = (long)off_hour*60*60 + off_minute*60 + off_second;
```

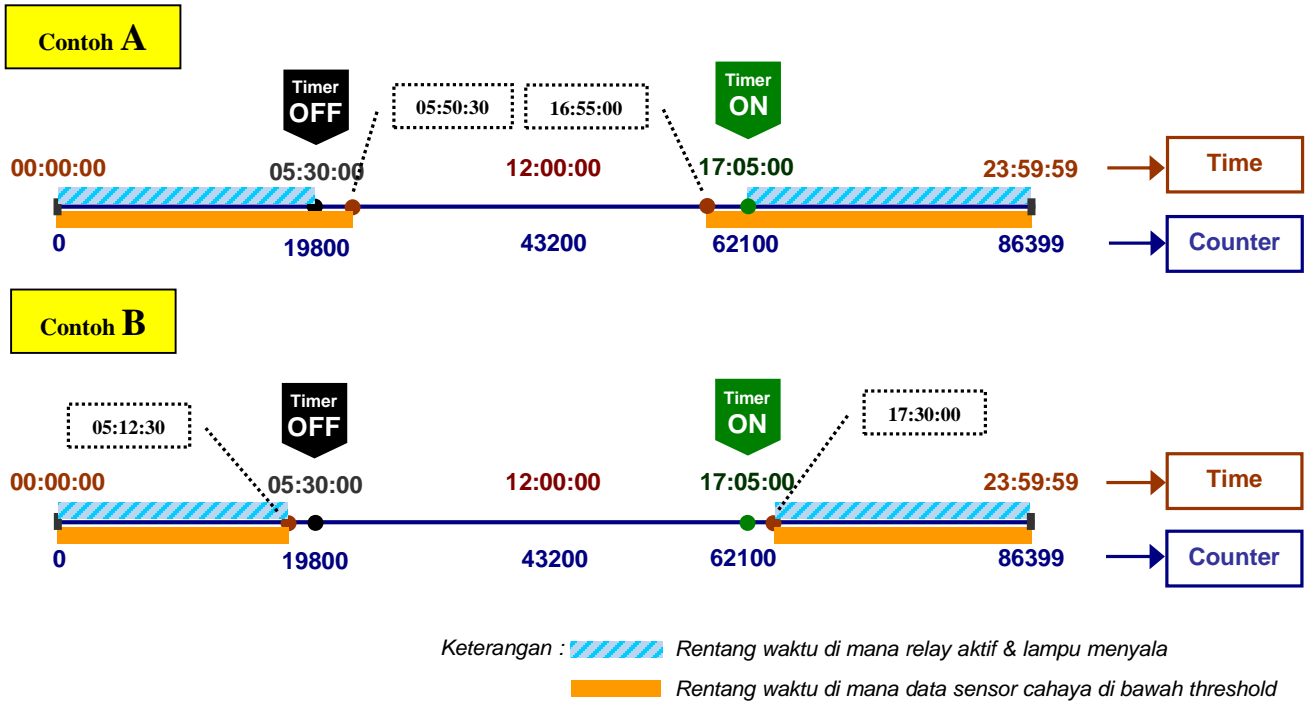
8. Baca data ADC sensor dari modul DT-Sense Light Sensor.

```
int value_lightSensor = analogRead(pin_lightSensor);
```

9. Kemudian berdasarkan data sensor dan data *counter* tersebut, akan dapat ditentukan kapan relay aktif dan nonaktif. Pada dasarnya hampir sama dengan percobaan kedua. Namun yang membedakan di sini adalah sensor cahaya dijadikan referensi pertama untuk menentukan apakah kondisi gelap atau terang. Selanjutnya dicocokkan dengan waktu/*timer* untuk memastikan bahwa pada saat tersebut memang benar-benar malam, bukan hanya mendung gelap saat siang hari. Dengan demikian akan diperoleh referensi yang lebih akurat untuk menentukan kapan relay harus diaktifkan dan kapan relay harus dinonaktifkan. Untuk lebih jelasnya dapat dilihat pada cuplikan kode program berikut ini.

```
if(value_lightSensor < threshold_lightSensor) {
    if(counter_OFF > counter_ON) {
        // jika nilai counter saat OFF > nilai counter saat ON,
        // misal ON pada jam 17:15 sore & OFF pada jam 21:45 malam harinya
        if(counter_NOW >= counter_ON && counter_NOW < counter_OFF) {
            digitalWrite(Relay1, HIGH); // Relay1 aktif sehingga Lampu nyala
            Serial.println("Nyala");
        }
        else {
            digitalWrite(Relay1, LOW); // Relay1 nonaktif sehingga Lampu padam
            Serial.println("Padam");
        }
    }
    else { // logikanya = if(counter_ON > counter_OFF)
        // jika nilai counter saat ON > nilai counter saat OFF,
        // misal ON pada jam 17:05 sore & OFF pada jam 05:30 keesokan paginya
        if(counter_NOW >= counter_OFF && counter_NOW < counter_ON) {
            digitalWrite(Relay1, LOW); // Relay1 nonaktif sehingga Lampu padam
            Serial.println("Padam");
        }
        else {
            digitalWrite(Relay1, HIGH); // Relay1 aktif sehingga Lampu nyala
            Serial.println("Nyala");
        }
    }
}
else
{
    digitalWrite(Relay1, LOW); // Relay1 nonaktif sehingga Lampu padam
    Serial.println("Padam");
}
```

Untuk lebih jelasnya mengenai cuplikan program tersebut, dapat dilihat pada ilustrasi berikut.



**Gambar 9**  
 Dua Contoh *Timeline* Ilustrasi Cara Kerja dari Percobaan Tiga

Seperti yang terlihat pada **Gambar 9**, dua contoh ilustrasi tersebut Timer ON/OFF diatur untuk menyalakan lampu pada pukul 17:05:00 kemudian memadamkannya pada pukul 05:30:00.

Pada **contoh A**, lampu dapat menyala dan mati sesuai jadwal, meski kondisi gelap (sensor cahaya di bawah *threshold*) terjadi sebelum Timer ON sampai setelah Timer OFF.

Namun pada **contoh B**, lampu baru menyala ketika pukul 17:30:00. Hal ini dikarenakan sensor cahaya dijadikan sebagai patokan/acuan pertama, baru kemudian RTC. Sederhananya, jika nilai sensor cahaya belum berada di bawah nilai *threshold*-nya, maka lampu tidak menyala sekalipun waktu RTC telah mencapai Timer ON. Sedangkan lampu padam pada 05:12:30 meski belum mencapai waktu Timer OFF dikarenakan sensor cahaya sudah berada di atas nilai *threshold*.

Demikian penjelasan aplikasi ini, tentunya masih banyak lagi yang bisa dikembangkan dari aplikasi dan dari modul **EMS Relay Shield** ini. Salah satunya jika ingin lebih *user friendly* maka bisa ditambahkan tombol sehingga konfigurasi waktu dapat diubah kapanpun tanpa memerlukan *upload* program baru.

**L**isting program aplikasi ini terdapat pada **AN230.ZIP**.

**S**elamat berinovasi!

*All trademarks, company names, product names and trade names are the property of their respective owners. All softwares are copyright by their respective creators and/or software publishers.*