

PROGRAMMABLE AUTOMATIC LIGHTING CONTROL SYSTEM

Benedictus Herry Suharto

Dosen Sekolah Tinggi Ilmu Komputer Yos Sudarso Purwokerto

Alamat e-mail: bherrys@gmail.com

ABSTRACT

The aim of this research is to design and implement a programmable automatic lighting control system. The system consists of microcontroller hardware that handle lighting to turn on or turn off based on schedule. The schedule made by software application on personal computer (PC). RTC (Real-Time Clock) was used to establish timing in real time world, in order to compare with time data has been programed. In addition, USB (universal Serial Bus) was used to establish communication between PC and the device, in order to program the schedule data in the device, so the lighting turn on or turn off schedule that runs can be customized by user.

The result of the research showed that the schedule in the device, as well as in the PC, was running well, leading us to the conclusion that the programmable automatic lighting control system can be used in real life.

Keywords : *schedule, automatic, lighting control, programmable, PC.*

1. PENDAHULUAN

Masyarakat konsumen listrik semakin sadar dalam mendukung program pemerintah untuk melakukan penghematan energi listrik. Pada sisi lain, kebutuhan untuk memberikan penerangan di sekitar fasilitas publik, seperti taman kota, area parkir stadion, area luar sekitar gedung bertingkat dan lain sebagainya merupakan sebuah keharusan. Sehingga pengelolaan hidup dan matinya lampu-lampu tersebut saat ini menjadi perhatian masyarakat.

Umumnya pengelola penerangan fasilitas publik tersebut tidak mempekerjakan karyawan khusus yang ditugasi untuk menghidupkan dan mematikan lampu-lampu. Tetapi menggunakan perangkat otomatis yang berupa sensor cahaya atau timer untuk menghidupkan dan mematikan lampu-lampu. Dengan menggunakan sensor cahaya, perangkat otomatis tersebut akan menyalakan lampu pada malam hari ketika tidak ada cahaya, dan mematikan lampu pada siang hari ketika cahaya menyentuh sensornya. Sedangkan perangkat otomatis berbasis timer akan menghidupkan dan mematikan lampu pada waktu tertentu ketika nilai timer yang di-set-kan terpenuhi.

Permasalahan yang terjadi adalah kadangkala perangkat otomatis tersebut tidak bekerja sesuai dengan fungsinya, sehingga memberikan kesan pemborosan energi listrik. Misalnya pada perangkat otomatis berbasis sensor cahaya, karena tertutup debu pada bagian sensornya atau tidak menerima cahaya yang cukup pada kondisi cuaca mendung, mengakibatkan lampu tidak mati pada siang hari. Pada perangkat otomatis berbasis timer, karena mesin pewaktuannya masih menggunakan teknologi mekanik, maka tingkat kepresisianya sangat rendah. Sehingga jika tidak di-set ulang secara berkala, akan mengakibatkan pergeseran waktu hidup dan matinya lampu. Hal ini menyebabkan lampu hidup atau mati tidak sesuai dengan kondisi yang ditetapkan, seperti hari sudah siang (pagi) lampu masih menyala atau masih terlalu siang (sore) tetapi lampu sudah menyala.

Tujuan yang ingin dicapai dalam penelitian ini adalah membuat dan mengimplementasikan suatu sistem pengendali lampu penerangan otomatis yang dapat diprogram ulang sesuai dengan jadwal yang telah ditentukan menggunakan perangkat PC (*personal computer*) atau *notebook*. Jadwal dikelola berdasarkan kebutuhan pengguna secara individual atau berkala, misalnya harian, mingguan, dan bulanan atau pada tanggal tertentu. Dalam penelitian ini digunakan perangkat keras *single chip computer* atau *chip* mikrokontroler AT89S52 yang berfungsi sebagai pengendali lampu otomatis dan perangkat keras PC yang digunakan sebagai antarmuka pengelolaan jadwal otomatisasi nyala lampu. Perangkat lunak untuk memprogram *chip* menggunakan bahasa pemrograman C, sedangkan perangkat lunak untuk memprogram antarmuka pengelolaan jadwal otomatisasi nyala lampu pada komputer PC menggunakan bahasa pemrograman Java.

Hasil penelitian diharapkan dapat digunakan oleh masyarakat luas sebagai alternatif implementasi untuk mengelola lampu-lampu penerangan fasilitas publik, seperti taman kota, area parkir stadion, area luar sekitar gedung bertingkat, area luar lingkungan kampus dan lain sebagainya. Sehingga nyala lampu-lampu penerangan tersebut dapat dikostumisasi sesuai dengan jadwal yang diinginkan oleh pengelolanya dalam rangka melakukan efisiensi penghematan energi listrik. Selain itu, penelitian ini dapat juga digunakan oleh peneliti lain yang membutuhkan perangkat pengendali terjadwal pada penelitiannya, seperti pada bidang industri, instrumentasi, peralatan navigasi, dan peralatan keamanan.

2. DASAR TEORI

2.1. Real-Time Clock

Real-Time Clock (RTC) adalah sebuah sistem pewaktuan komputer yang digunakan untuk menjaga waktu suatu perangkat elektronik agar tetap sejalan atau sesuai dengan kondisi waktu nyata (----, 2010). Dalam kebanyakan aplikasi elektronik, RTC muncul dalam bentuk

rangkaian terpadu (*integrated circuit* atau IC). Istilah *Real-Time Clock* digunakan untuk menghindari penafsiran yang salah dengan istilah “*hardware clock*” yang hanya berupa sinyal pewaktuan untuk mensinkronkan komponen-komponen digital dalam sebuah rangkaian digital dan bukan merupakan unit pengukuran waktu dalam kehidupan manusia.

Beberapa keuntungan berikut ini dapat diperoleh dengan mengimplementasikan RTC untuk pengelolaan waktu:

- Mengkonsumsi daya yang sangat rendah (diperlukan ketika catu daya utama mati)
- Membebaskan sistem utama dari tugas menyediakan pewaktuan kritis.
- Akurasi tinggi, karena bukan merupakan bagian dari sistem.

RTC memiliki *backup* catu daya sendiri agar ketika catu daya utama terputus, penghitungan waktu nyata tetap terjaga. Umumnya *backup* catu daya ini berupa baterai *lithium* yang memiliki dimensi bentuk relatif kecil, sehingga dapat diimplementasikan pada perangkat *portable*.

Umumnya RTC menggunakan osilator kristal untuk menggerakkan pewaktuannya. Osilator ini sama seperti osilator yang digunakan pada jam digital dan *quartz clock*. Besar frekuensi yang dihasilkan oleh osilator adalah 32.768 kHz yang tepat memiliki siklus 2^{15} per detik sesuai untuk ditempatkan pada pencacah sederhana.

2.2. Sistem kendali

Sistem Kendali adalah suatu sistem yang bertujuan untuk mengendalikan suatu proses agar *output* yang dihasilkan dapat dikontrol sehingga tidak terjadi kesalahan (-----, 2010). Dalam hal ini *output* yang dikendalikan adalah kestabilannya, ketelitian, dan kedinamisannya. Secara umum, sistem kendali dapat dibedakan menjadi dua jenis yaitu sistem kendali *loop* terbuka dan *loop* tertutup.

Sistem Kendali *loop* terbuka, keluaranya tidak mempengaruhi *input*. Atau dengan kata lain sistem kendali *loop* terbuka keluarannya (*output*) tidak dapat digunakan sebagai perbandingan umpan balik dengan *input*-nya. Akibatnya ketetapan dari sistem tergantung dari kalibrasi. Pada umumnya, sistem kendali *loop* terbuka tidak tahan terhadap gangguan luar. Sedangkan pada sistem kendali *loop* tertutup, sinyal kesalahan yang bekerja, yaitu perbedaan antara sinyal *input* dan sinyal umpan balik *diinputkan* kekontroller sedemikian rupa untuk mengurangi kesalahan dan membawa keluaran sistem kenilai yang dikehendaki. Pada umumnya sistem kendali *loop* tertutup tahan terhadap gangguan dari luar.

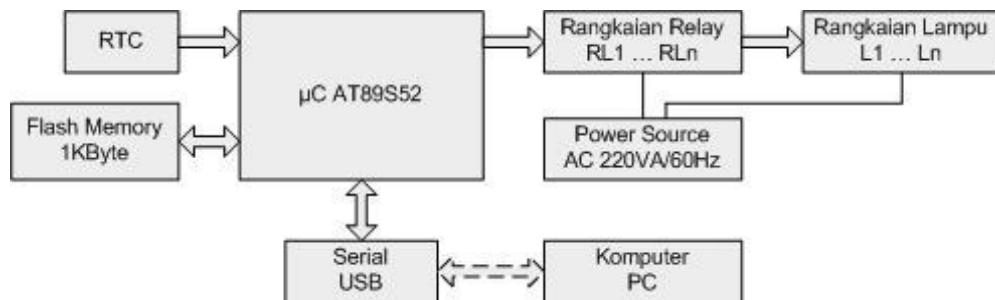
Secara garis besar, sistem kendali jika ditinjau dari ketelitian dan kestabilan sistem dapat dibagi atas dua bagian, yaitu sistem kendali dengan menggunakan PID kontroller dan sistem kendali *on-off*. Khusus pada sistem kendali *on-off* ada dua keadaan yang akan dihasilkan *output* yaitu keadaan *on* atau keadaan *off*. Mikroprosesor dapat digunakan sebagai pengendali sistem kendali *on-off*.

3. METODE PENELITIAN

Metodologi yang dilakukan dalam penelitian ini adalah sebagai berikut:

- 1) Merancang dan menguji perangkat keras pengendali lampu otomatis.
- 2) Merancang perangkat lunak pengendali lampu otomatis.
- 3) Merancang perangkat lunak pengelolaan jadwal otomatisasi nyala lampu.
- 4) Menganalisa hasil.
- 5) Membuat kesimpulan.

3.1. Perancangan dan Pengujian Perangkat Keras Pengendali Lampu Otomatis



Gambar 1. Diagram kotak sistem pengendali lampu otomatis

RTC digunakan sebagai referensi waktu bagi sistem pengendali lampu otomatis. Ketika pertama kali perangkat pengendali lampu otomatis dijalankan, pewaktuan pada perangkat ini akan diinisialisasi dengan nilai yang diambil dari RTC. Nilai waktu inilah yang nantinya digunakan sebagai pembanding untuk mengambil keputusan apakah sebuah lampu menyala atau mati. Untuk menyimpan data jadwal operasi seluruh lampu yang dikirim dari PC digunakan memori Flash (*nonvolatile*) sebesar 1024 byte. Data dalam memori ini, satu per satu dibandingkan dengan nilai waktu yang berasal dari RTC setiap satu menit. Jika data sebuah lampu sesuai dengan nilai yang dibandingkan, maka mikrokontroller akan mengaktifkan *relay* untuk menghidupkan lampu bersangkutan.

Mikrokontroller AT89S52 hanya memiliki *port* komunikasi serial dengan level logika TTL (*Transistor–Transistor Logic*). Agar dapat berkomunikasi langsung dengan PC menggunakan *port* USB (*Universal Serial Bus*) dan protokol RS232, perlu ditambahkan

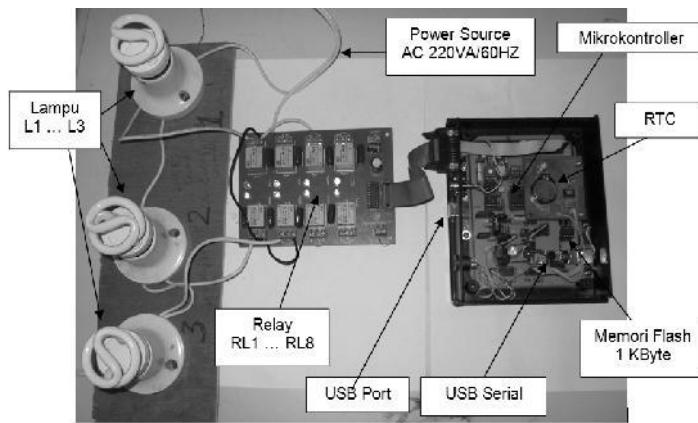
Diagram kotak dari sistem perangkat keras pengendali lampu otomatis adalah sebagai berikut:



Gambar 2. Modifikasi kabel data Nokia CA-42.

Rangkaian *relay* yang dipakai dalam penelitian ini terdiri dari 8 *relay* individual yang langsung dikendalikan dari *port* 0 (8 jalur) mikrokontroller melalui sebuah *buffer* penggerak *relay*. Ketika sebuah jalur *port* 0 memiliki nilai logika 1, maka *buffer* penggerak akan mengaktifkan *relay* bersangkutan untuk menyalaikan lampu.

Berikut ini adalah gambar perangkat keras sistem pengendali lampu otomatis yang telah dibuat:

**Gambar 3.** Perangkat keras sistem pengendali lampu otomatis.

Perangkat keras komputer PC yang digunakan dalam penelitian ini adalah perangkat *notebook* yang memiliki spesifikasi sebagai berikut:

- Prosessor Intel Centrino 1,4 GHz.
- Memory 1024 MByte
- Harddisk 40 GByte
- I/O : 2 port USB 2.1, 1 port Audio Out, 1 port Mic.
- Sistem Operasi Windows XP SP3

Pengujian perangkat keras sistem pengendali lampu otomatis dilakukan dengan menggunakan program sederhana untuk mengaktifkan *relay* (lampu) secara bergantian

dengan selang 1 menit, 2 menit, 4 menit, dan 8 menit. Pada menit ke 8 seluruh *relay* diaktifkan. Pada program pengujian ini, RTC di-set terlebih dahulu dengan nilai waktu nyata saat pengujian dilakukan.

Untuk memastikan bahwa perangkat sistem pengendali lampu otomatis telah berfungsi sesuai dengan rancangan, pengujian dilakukan dengan cara pengamatan langsung. Sistem diaktifkan dan kondisi lampu diamati apakah telah sesuai dengan keadaan yang telah dikondisi dalam pengujian, yaitu lampu menyala bergantian dengan selang waktu 1, 2, dan 4 menit serta menyala semua pada menit ke 8. Proses pengujian dapat dilihat pada gambar 4.

**Gambar 4.** Pengujian perangkat keras sistem pengendali lampu otomatis.

Pengujian komunikasi serial USB antara perangkat sistem pengendali lampu otomatis dengan komputer PC dilakukan dengan

menambahkan rutin program pengujian komunikasi serial RS232 (UART) pada perangkat sistem pengendali lampu otomatis.

Rutin program ini berisi konfigurasi parameter komunikasi (*baudrate*, *data bit*, *parity*, dan *stop bit*) dan data-data yang akan dikomunikasikan pada saat pengujian. Prosedur rutin program yang sama juga diterapkan pada program pengujian di sisi komputer PC. Pengujian ini dikatakan telah berhasil jika komputer PC mengirimkan data integer “123”, maka akan dijawab oleh sistem pengendali lampu otomatis dengan menyalakan lampu L3. Jika komputer PC mengirimkan data integer “456” maka akan dijawab dengan mengirimkan nilai data waktu yang ada dalam RTC ke komputer PC.

3.2. Perancangan dan Pengujian Perangkat Lunak Pengendali Lampu Otomatis

Perangkat lunak sistem pengendali lampu otomatis yang dirancang berisi 4 bagian algoritma pemrograman, yaitu:

- Algoritma bagian inisialisasi perangkat pengendali lampu otomatis yang berfungsi untuk mengkonfigurasi protokol akses flash memori, protokol akses RTC, protokol komunikasi serial, jalur *port* penggerak *buffer relay*, dan variabel-variabel pendukung program.
- Algoritma bagian kendali lampu yang berfungsi membandingkan data RTC dengan data dalam memori Flash yang hasilnya digunakan untuk menggerakkan *buffer relay* (menyalakan lampu).
- Algoritma bagian komunikasi serial USB yang berfungsi untuk menerima dan mengirim data jadwal nyala lampu dari dan ke komputer PC.
- Algoritma bagian pengaksesan memori yang berfungsi mengelola pembacaan dan penulisan data jadwal nyala lampu dari dan ke flash memori.

Pada kondisi normal (*running*), program akan membandingkan data waktu yang berada dalam RTC dengan data waktu yang berada dalam

memori Flash secara berkala dengan interval 1 menit. Algoritma proses

pembandingan data waktu ini memiliki 4 prioritas urutan sebagai berikut:

- 1) Membandingkan data jadwal tanggal tertentu. Pertama program melihat data tanggal saat ini dari RTC. Apabila pada tanggal sekarang sama dengan tanggal yang di simpan pada memori Flash, maka program membandingkan data jam. Jika data jam sama, maka akan dilanjutkan dengan membandingkan data menit. Jika data menit sama, maka program akan membaca data *relay* (lampu) yang akan diaktifkan. Program kemudian akan melihat dan membandingkan isi *counter* (pencacah) lampu bersangkutan (CL_n) dengan nilai lama waktu untuk nyala lampu yang ditetapkan pada data jadwal. Jika nilainya lebih kecil, maka data *counter* akan diganti dengan nilai lama waktu nyala lampu. Sebaliknya, jika nilainya lebih besar, maka nilai *counter* tidak diubah dan data nilai lama waktu nyala lampu diabaikan. Proses pengubahan nilai *counter* CL_n dilakukan agar tidak mengganggu penetapan jadwal lain (jadwal harian, harian dan bulanan) pada lampu bersangkutan.
- 2) Membandingkan data jadwal bulanan. Algoritma pada prioritas ini mirip dengan prioritas pertama, tetapi data yang dibandingkan adalah data jadwal bulanan pada memori Flash dengan data saat ini pada RTC.
- 3) Membandingkan data jadwal mingguan. Algoritma pada prioritas ini mirip dengan prioritas pertama, tetapi data yang dibandingkan adalah data jadwal mingguan pada memori Flash dengan data saat ini pada RTC.
- 4) Membandingkan data jadwal harian. Algoritma pada prioritas ini mirip dengan prioritas pertama, tetapi data yang dibandingkan adalah data jadwal harian pada memori Flash dengan data saat ini pada RTC.

Proses selanjutnya adalah mengurangi 1 (satu) nilai *counter* CL_n setiap interval 1 menit untuk menghitung lama nyala lampu L_n . Ketika nilai $CL_n = 0$, program akan menonaktifkan *relay* (lampu L_n).

Pada kondisi program, komunikasi serial perangkat sistem pengendali lampu otomatis diinisialisasi menggunakan mode 1 yang merupakan mode 8-bit UART. Pada mode komunikasi ini dibutuhkan pembangkit *baud rate* yang dihasilkan oleh timer 1 dalam mode 2. Untuk memperoleh *baud rate* sebesar 9600 bps (*bit per second*), SFR TH1 diberi nilai dengan menggunakan formula berikut (Schultz, 2004):

$$\text{Baud rate} = \frac{(K * \text{Frekuensi osilator})}{(32 * 12 * (256 - TH1))}$$

$$9600 = \frac{(1 * 11.059.000)}{(32 * 12 * (256 - TH1))} \quad \Rightarrow K = 1 \text{ untuk SMOD} = 0$$

TH1 = 253 atau 0xFD (dalam heksadesimal)

Ketika data jadwal nyala lampu diterima melalui *port* serial, program akan memanggil rutin algoritma penyimpanan data ke dalam memori flash melalui port P3.6 (SCL) dan P3.7 (SDA) menggunakan protokol I²C (*Inter-Integrated Circuit*). Gambar 5 memperlihatkan format protokol I²C, sedangkan langkah penulisan dan pembacaan data pada memori flash menggunakan protokol I²C (Parab, 2008) (Durham, 2004) adalah sebagai berikut:

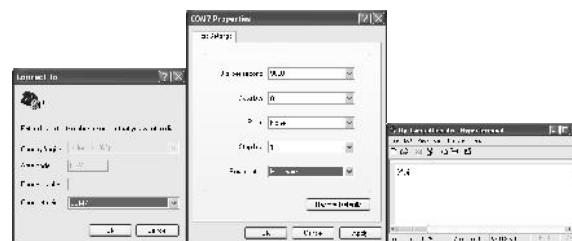
- 1) Memanggil rutin Start() untuk mengirim bit START.
- 2) Memanggil rutin WriteI2C(alamat) untuk menunjuk pada alamat yang dituju.
- 3) Memanggil rutin WriteI2C(baca/tulis) untuk mengirim bit baca atau bit tulis.
- 4) Memanggil rutin ReadI2C atau WriteI2C untuk mendapatkan atau mengirim bit acknowledge.
- 5) Memanggil rutin ReadI2C atau WriteI2C untuk membaca atau menulis byte data.
- 6) Memanggil rutin ReadI2C atau WriteI2C untuk mendapatkan atau mengirim bit acknowledge.

- 7) Memanggil rutin Stop() untuk mengirim bit STOP.



Gambar 5. Format Protokol I²C. Perangkat

lunak sistem pengendali lampu otomatis diuji dengan mengirimkan simulasi data jadwal nyala lampu harian, mingguan, bulanan dan tanggal tertentu ke perangkat sistem pengendali lampu otomatis menggunakan aplikasi “Hyper Terminal” (gambar 6) yang ada pada sistem operasi Windows XP.



Gambar 6. Aplikasi “Hyper Terminal” dan konfigurasi yang dipakai dalam pengujian.

Agar dapat berkomunikasi dengan perangkat sistem pengendali lampu otomatis, Hyper terminal perlu dikonfigurasi dengan parameter seperti pada gambar 6. Data simulasi yang dikirim ke perangkat sistem pengendali lampu otomatis dapat dilihat pada tabel 1. Ketika seluruh data simulasi telah dikirimkan ke perangkat sistem pengendali lampu otomatis, proses pengujian dilanjutkan dengan mengamati kondisi lampu-lampu apakah telah sesuai dengan pengkondisian yang diterapkan pada simulasi data jadwal nyala lampu harian, mingguan, bulanan dan tanggal tertentu.

Tabel 1. Data simulai yang dikirim ke perangkat sistem pengendali lampu otomatis.

| Kode | M | J | H | G | B | T | C | L | Keterangan |
|------|----|----|----|----|----|------|----|----|--|
| FE | 00 | 11 | 00 | 00 | 00 | 0000 | 3C | 01 | L1 menyala setiap hari pukul 17:00 selama 1 jam |
| FE | 0F | 11 | 00 | 00 | 00 | 0000 | 1E | 06 | L2 & L3 menyala setiap hari pukul 17:30 selama 30 menit |
| FD | 00 | 13 | 01 | 00 | 00 | 0000 | 3C | 01 | L1 menyala setiap minggu pukul 19:00 selama 1 jam |
| FD | 0F | 13 | 01 | 00 | 00 | 0000 | 1E | 06 | L2 & L3 menyala setiap Minggu pukul 19:30 selama 30 menit |
| FC | 00 | 15 | 00 | 15 | 00 | 0000 | 3C | 01 | L1 menyala setiap tanggal 21 pukul 21:00 selama 1 jam |
| FC | 0F | 15 | 00 | 15 | 00 | 0000 | 1E | 06 | L2 & L3 menyala setiap tanggal 21 pukul 21:30 selama 30 menit |
| FB | 00 | 17 | 00 | 15 | 03 | 07DA | 3C | 01 | L1 menyala tanggal 21 Maret 2010 pukul 23:00 selama 1 jam |
| FB | 0F | 00 | 00 | 16 | 03 | 07DA | 1E | 06 | L2 & L3 menyala tanggal 22 Maret 2010 pukul 0:30 selama 30 menit |

Keterangan tabel:

- Kode kelompok data jadwal adalah FE = harian, FD = mingguan, FC = bulanan, FB = tanggal tertentu.
- Data jadwal adalah M = menit, J = jam, H = nama hari (1 = Minggu, 2 = Senin, ... dst), G = tanggal, B = bulan (1 = Januari, 2 = Februari, ... dst) , T = tahun, C = nilai counter, dan L data lampu dalam biner (00000001 = L1, 00000110 = L1 dan L2).

3.3. Perancangan dan Pengujian Perangkat Lunak pengelolaan jadwal

3.3.1. Kebutuhan Sistem

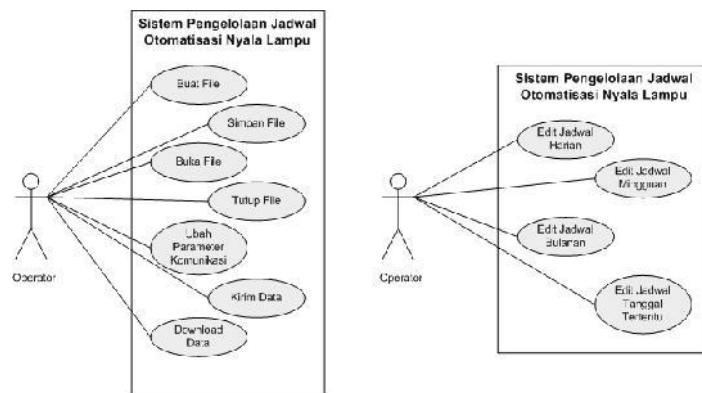
Ada tiga kebutuhan yang harus disediakan oleh sistem aplikasi Pengelolaan Jadwal Otomatisasi Nyala Lampu, yaitu:

- 1) Kemampuan untuk mengolah data jadwal nyala lampu.
- 2) Kemampuan untuk menyimpan dan menggunakan kembali data jadwal nyala lampu yang telah diolah.
- 3) Kemampuan untuk melakukan komunikasi ke perangkat pengendali

lampu otomatis, sekaligus dapat mengirim dan mengambil data jadwal nyala lampu ke dan dari pengendali lampu otomatis.

3.3.2. Perancangan Sistem

Sistem aplikasi Pengelolaan Jadwal Otomatisasi Nyala Lampu merupakan sebuah aplikasi *stand alone*, oleh karenanya pada sistem aplikasi ini hanya memiliki satu aktor, yaitu operator. Operasi yang dapat dilakukan oleh aktor operator digambarkan pada diagram *use case* berikut ini:

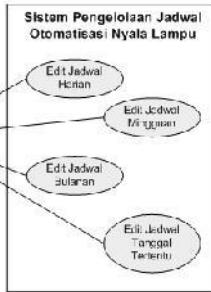
**Gambar 7.** Use case Sistem Pengolahan Jadwal Otomatisasi Nyala Lampu.

3.3.3. Perancangan Antarmuka Pengguna Sistem

Desain antarmuka sistem aplikasi Pengelolaan Jadwal Otomatisasi Nyala Lampu dirancang berbasis pada GUI Swing Java. Antarmuka utama berupa daftar tabel-tabel jadwal nyala lampu yang dikelompokkan dalam harian, mingguan (Minggu, Senin, Sabtu), bulanan (tanggal 1, 2, 3, ... 31), dan tanggal tertentu. Pengguna dapat melakukan interaksi dengan sistem melalui menu-menu navigasi dan tombol-tombol fungsi. Menu utama terdiri atas empat menu, yaitu:

- 1) Berkas, berisi sub menu Baru, Buka, Simpan, Tutup, dan Keluar.
- 2) Edit, berisi sub menu Konfig COM, Uji COM, Set Waktu.
- 3) Data, berisi sub menu Jadwal Harian, Jadwal Mingguan, Jadwal Bulanan, Jadwal perTanggal, Kirim Data, Download Data.
- 4) Bantuan, berisi sub menu Tentang Program, Lisensi dan Manual.

Berikut ini disain antarmuka yang dirancang:

**Gambar 8.** Disain antarmuka aplikasi

3.3.4. Perancangan Program

Bahasa pemrograman Java menyediakan class GUI swing yang lengkap, sehingga mempermudah pembuatan program Aplikasi Pengelolaan Jadwal Otomatisasi Nyala Lampu. Beberapa komponen Swing “siap-pakai” yang dipakai dalam pemrograman aplikasi ini adalah *JFrame*, *JTable*, *JTabbedPane*, *JMenu*, *JMenuItem*, *JButton*, *JTextField*, *JDialog*, *JFileChooser*, dan *JPanel*. Program aplikasi bantu pemrograman yang digunakan adalah IDE Netbeans 5.5. Dengan menggunakan aplikasi bantu ini, programmer hanya perlu melakukan pengubahan dan penyesuaian properti komponen-komponen swing untuk membangun aplikasi sesuai dengan rancangan antarmuka pengguna yang telah dibuat (Hartati, 2007).

Komponen API Java yang memiliki peranan penting dalam pemrograman ini adalah class *InputStream*, *OutputStream*, *FileInputStream*, dan *FileOutputStream*. Class-class ini digunakan untuk mengalirkan data keluar dan masuk program. Sedangkan pustaka eksternal yang dipakai adalah pustaka *rxtx* yang dirancang oleh Keane Jarvi (www.rxtx.org). Pustaka ini digunakan untuk melakukan komunikasi dengan perangkat sistem pengendali lampu otomatis menggunakan protokol RS232. Berikut ini method *setSerialPortParameters()* dan *connect()* yang digunakan dalam program untuk melakukan komunikasi dengan perangkat sistem pengendali lampu otomatis:


```

private void setSerialPortParameters() throws
IOException {
    int baudRate = 9600;
    try {
        serialPort.setSerialPortParams(
            baudRate,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);

        serialPort.setFlowControlMode(SerialPort.FLOWCO
NTROL_NONE);
    } catch (UnsupportedCommOperationException ex) {
    }

    public boolean connect(String portName) throws
IOException {
        boolean sukses = true;
        try {
            portId =
            CommPortIdentifier.getPortIdentifier(portName);
            serialPort = (SerialPort) portId.open("KLO test",
            5000);
            setSerialPortParameters();
            outputStream = serialPort.getOutputStream();
        }
    }
}

```

```

inputStream = serialPort.getInputStream();
} catch (NoSuchPortException ex) { sukses = false;
} catch (PortInUseException e) { sukses = false;
} catch (IOException e) { serialPort.close(); sukses =
false; }
return sukses;
}

```

4. HASIL PENELITIAN DAN PEMBAHASAN

Data hasil pengujian menggunakan aplikasi “Hyper Terminal” dan pengamatan unjuk kerja perangkat pengendali lampu otomatis dapat dilihat pada tabel 2. Untuk mengukur ketepatan waktu digunakan jam digital dan *stopwatch* buatan pabrik. Selisih antara waktu lampu menyala dan mati dengan waktu yang ditampilkan pada jam digital dan *stopwatch* sangat kecil. Selisih waktu ini muncul karena faktor kesalahan manusia yang diakibatkan dari keterlambatan tanggapan antara pengamatan saat kondisi lampu berubah dan penekanan tombol “*stop*” dari jam digital serta *stopwatch*. Kesalahan ini kemudian diabaikan, karena setelah dilakukan pengulangan berkali-kali (3 sampai 5 kali), hasil yang diperoleh tetap menunjukkan kesalahan yang sama, yaitu selisih waktu dengan besaran yang mirip tetapi acak pada setiap kejadian kondisi perubahan lampu. Dengan demikian, dapat disimpulkan bahwa perangkat yang dibuat telah sesuai dengan rancangan.

Tabel 2. Data pengujian dan pengamatan kerja perangkat pengendali lampu otomatis.

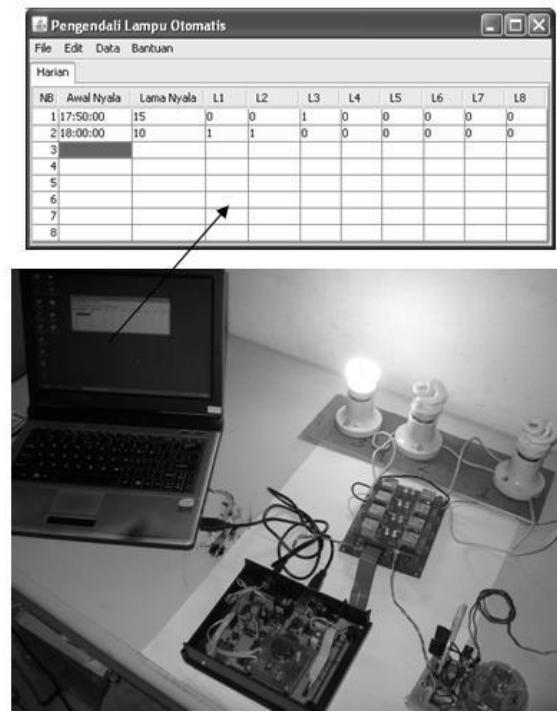
| Data Jadwal | | | Data Pengamatan | | | Selisih |
|-------------|----------|-------------|-----------------|----------|--------------|---------|
| Hidup | Mati | Durasi | Hidup | Mati | Durasi | |
| 17:00:00 | 18:00:00 | 60 mnt 0 dt | 16:59:59 | 18:00:01 | 60 mnt 2dt | 2 dt |
| 17:30:00 | 18:00:00 | 30 mnt 0 dt | 17:30:00 | 18:00:02 | 30 mnt 2 dt | 2 dt |
| 19:00:00 | 20:00:00 | 60 mnt 0 dt | 19:00:01 | 20:00:00 | 59 mnt 59 dt | -1 dt |
| 19:30:00 | 20:00:00 | 30 mnt 0 dt | 19:30:00 | 20:00:01 | 30 mnt 1 dt | 1 dt |

| | | | | | | |
|----------|----------|-------------|----------|----------|-------------|------|
| 21:00:00 | 22:00:00 | 60 mnt 0 dt | 21:00:00 | 22:00:01 | 60 mnt 1 dt | 1 dt |
| 21:30:00 | 22:00:00 | 30 mnt 0 dt | 21:30:01 | 22:00:00 | 30 mnt 1 dt | 1 dt |
| 23:00:00 | 00:00:00 | 60 mnt 0 dt | 23:00:00 | 00:00:00 | 60 mnt 0 dt | 0 dt |
| 23:00:00 | 00:00:00 | 30 mnt 0 dt | 23:00:01 | 00:00:02 | 30 mnt 1 dt | 1 dt |

Pengujian keseluruhan sistem (pengendali lampu otomatis dengan PC) dilakukan dengan membuat atau menulis data jadwal nyala lampu menggunakan aplikasi sistem pengelolaan jadwal. Data ini kemudian dikirimkan ke perangkat pengendali lampu otomatis menggunakan koneksi USB. Hasil pengamatan memperlihatkan bahwa setelah seluruh proses pengiriman selesai dan perangkat pengendali lampu otomatis berada pada kondisi *running*, perangkat tersebut dapat mengontrol

penyalaan lampu yang dikendalikannya. Lampu menyala dan mati sesuai dengan data jadwal yang dibuat. Pada proses pengambilan (*download*) data dari perangkat pengendali lampu otomatis ke PC, juga memperlihatkan konsistensi data yang sama seperti pada data saat dikirim ke perangkat.

Konfigurasi perangkat keras dan cuplikan aplikasi pada saat pengujian ini dapat dilihat pada gambar 9.



Gambar 9. Konfigurasi saat pengujian.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian dan implementasi pada penelitian ini, maka dapat ditarik kesimpulan sebagai berikut:

1. Perangkat sistem pengendali lampu otomatis yang dirancang dapat diprogram ulang (*programmable*) dengan menggunakan PC atau notebook untuk menkostumisasi jadwal penyalaan lampu-lampu yang dikelolanya.
2. Aplikasi sistem pengelolaan data jadwal otomatisasi nyala lampu yang dirancang memberikan kemudahan dalam penyuntingan jadwal penyalaan lampu dan proses pemrograman ulang data jadwal penyalaan lampu pada perangkat pengendali lampu otomatis.
3. Nilai tambah dari sistem aplikasi yang dikembangkan adalah kemampuan untuk dijalankan pada berbagai sistem operasi (*multiplatform*), seperti Linux dan MacOS. Hal ini dimungkinkan karena aplikasi dibuat menggunakan bahasa pemrograman Java yang *multiplatform*.

Berikut ini beberapa saran yang dapat dilakukan untuk menyempurnakan sistem yang telah dibuat:

1. Perangkat keras pengendali lampu otomatis dapat dikembangkan untuk menangani lebih banyak lampu dengan menambah komponen perluasan I/O, seperti PPI 8255. PPI 8255 memiliki 32 jalur port keluaran, sehingga dengan menambahkan komponen ini akan dapat mengendalikan relay (lampu) sebanyak 32 buah dan berlaku pula untuk kelipatannya.
2. Perangkat keras dan lunak sistem pengendali lampu otomatis dapat ditingkatkan kemampuannya dengan menambahkan fungsi deteksi nyala lampu. Dengan penambahan fungsi ini, keberadaan kondisi lampu apakah telah menyala atau belum dapat diketahui pada saat relay lampu

diaktifkan oleh perangkat pengendali lampu otomatis.

6. DAFTAR PUSTAKA

- 2010. *Real-time clock*, Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Real-time_clock. diakses Maret 2010.
- 2010. *Control Theory*, Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Control_theory. diakses Maret 2010.
- Schultz, T. 2004, *C and the 8051*, PageFree Publishing.
- Parab, Jivan S. Shinde, Santosh A. Shelake, Vinod G. Kamat, Rajanish K. Naik, Gourish M. 2008. *Practical Aspects of Embedded System Design using Microcontrollers*, Springer Science.
- Durham, Marcus O. 2004, *Systems Design and the 8051: The hardware, firmware, and software design of microprocessor systems*, 2nd. ed. TechnoPress, Tulsa.
- Hartati, G. Sri. Suharto, B. Herry. Susilo, M. (2007). *Pemrograman GUI Swing Java dengan Netbeans 5*, 1st. Ed. ANDI Yogyakarta

JURNAL MEDIA APLIKOM

ISSN : 2086 – 972X

Vol 1 No.2 Mei 2010