

# Perancangan Aplikasi Timer Elektronik Dengan Arduino Kit Untuk Mengendalikan Rangkaian Relay

Stepanus<sup>1</sup>, Riyadi Jimmy Iskandar<sup>2</sup>, Tony Darmanto<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, STMIK Widya Dharma, Pontianak

e-mail: <sup>1</sup>[ah.khiang2@gmail.com](mailto:ah.khiang2@gmail.com), <sup>2</sup>[riyadi@stmik-widyadharma.ac.id](mailto:riyadi@stmik-widyadharma.ac.id), <sup>3</sup>[tony@stmik-widyadharma.ac.id](mailto:tony@stmik-widyadharma.ac.id)

## Abstract

*The progress of digital electronics have resulted in the various form of ready used-hardware which includes devices like microcontroller circuits. By implementing the digital electronic techniques combined with the microcontroller, it is possible to design a tool with certain and specific automatization. The existence of automatization is essential to support human works. This research produces a timer application in computer and hardware which represented in a relay circuit connected to Arduino's microcontroller. The user just need to set the function and activation time interval by using the application's interface in computer, then the relay will disconnected-connected the electric current based on the timer-controlled microcontroller's output. After this research has accomplished, it can be concluded that the computer-controlled timer is suitable to be used in a repetitive work-scheduling without a necessity of high time accuracy. The maximum power in care of the relay must be concerned (without exceeds the relay's maximum power). The suggestion of the writer for further development of this research related with the combination of database utilization for saving the setting of function and time interval worked out by the user, so when the electric current 'down', the user don't need to repeat the setting.*

**Keywords**—Arduino, Microcontroller, Relay, Timer

## Abstrak

Perkembangan elektronika digital menghasilkan beragambentuk perangkat keras siap pakai yang mencakup perangkat-perangkat seperti rangkaian mikrokontroler. Dengan menggunakan teknik elektronika digital yang dipadukan dengan mikrokontroler dimungkinkan untuk merancang suatu alat dengan otomatisasi tertentu yang bersifat spesifik. Keberadaan otomatisasi cukup penting untuk meringankan beban kerja manusia dalam melakukan suatu pekerjaan. Penelitian ini menghasilkan suatu aplikasi *timer* pada komputer dan perangkat keras berupa rangkaian *relay* yang dihubungkan ke mikrokontroler Arduino. Pengguna cukup melakukan pengaturan fungsi dan rentang waktu pengaktifan melalui antarmuka aplikasi di komputer, selanjutnya *relay* akan memutus dan menyambungkan arus listrik pada beban sesuai *output* mikrokontroler yang dikendalikan oleh aplikasi *timer*. Setelah penelitian dilakukan, dapat disimpulkan bahwa *timer* yang dikontrol oleh komputer cocok untuk digunakan pada penjadwalan proses kerja yang berulang tanpa memerlukan akurasi waktu yang tinggi. Daya maksimum yang dibebankan kepada *relay* harus diperhatikan (tanpa melewati daya maksimum *relay*). Saran penulis untuk penelitian lanjutan adalah pengkombinasian penggunaan *database* pada aplikasi untuk menyimpan pengaturan fungsi dan rentang waktu yang diatur pengguna sehingga pada saat terjadi kegagalan arus listrik, pengguna tidak perlu melakukan pengaturan ulang.

**Kata kunci**—Arduino, Mikrokontroler, Relay, Timer

## 1. PENDAHULUAN

Dewasa ini teknologi informasi mengalami perkembangan yang sangat pesat, sehingga memberikan keuntungan dan kemudahan tersendiri bagi manusia *modern* untuk menunjang kehidupannya. Satu di antaranya yaitu perkembangan elektronika digital yang banyak digunakan dalam berbagai produk elektronik *modern* saat ini, seperti komputer, televisi, *handphone* dan produk-produk lainnya.

Perkembangan elektronika digital juga menghasilkan berbagai produk perangkat keras siap pakai yang terdiri dari rangkaian mikrokontroler, sehingga memberikan kemudahan kepada pengguna dalam hal

mempelajari pemrograman dan memanfaatkan mikrokontroler (tidak harus merakitnya dari awal). Beberapa contoh produk tersebut adalah Arduino, Freeduino, dan Zigduino.

Penggunaan paduan mikrokontroler dan teknik tertentu pada elektronika digital memungkinkan manusia merancang suatu alat dengan otomatisasi tertentu yang bersifat spesifik, sehingga memberikan sejumlah keuntungan bagi kehidupan manusia, karena otomatisasi akan mengurangi ketergantungan, campur tangan dan meringankan beban kerja manusia dalam melakukan suatu pekerjaan.

Dalam dunia komputer, otomatisasi dapat diwujudkan menggunakan bahasa pemrograman sebagai sarana komunikasi ke perangkat mikrokontroler yang akan dikendalikan. Pada awalnya, bahasa yang dimengerti mikrokontroler hanyalah berupa bahasa mesin, namun seiring perkembangan bahasa pemrograman dan antarmuka perangkat keras itu sendiri, saat ini mikrokontroler juga dapat dikendalikan atau diprogram menggunakan bahasa pemrograman tingkat tinggi seperti bahasa C ataupun bahasa *Basic* tanpa harus mempelajari bahasa mesin terlebih dahulu.

Berdasarkan uraian tersebut, penulis terdorong untuk merancang suatu aplikasi komputer sederhana yang berfungsi sebagai pewaktu elektronik. Dengan adanya pewaktu elektronik yang dibuat dengan bahasa pemrograman, maka mikrokontroler dapat difungsikan untuk mengendalikan rangkaian *relay* yang pada akhirnya menjadi saklar elektronik otomatis sehingga dapat membantu mengurangi beban kerja dan campur tangan manusia dalam melakukan suatu pekerjaan. Dengan demikian, masalah yang akan dibahas dalam penelitian ini berhubungan dengan cara penulis merancang suatu aplikasi *timer* elektronik menggunakan Arduino Kit untuk mengendalikan rangkaian *relay*. Penelitian ini tidak membahas komponen pembangun mikrokontroler yang digunakan secara spesifik. Aplikasi yang dirancang dapat berfungsi selama terhubung ke Arduino Kit pada komputer yang ter-*install* sistem operasi Windows, dan hanya memiliki ketepatan jam dan menit. Jumlah *relay* yang dapat dikendalikan terbatas sebanyak lima unit.

## 2. METODE PENELITIAN

### 3.1. Rancangan Penelitian, Teknik Analisis dan Perancangan Sistem

#### 2.1.1 Rancangan Penelitian

Penulis menggunakan Desain Penelitian Hubungan Kausal (eksperimental), yaitu melakukan percobaan dan pengujian terhadap aplikasi yang dibuat dan mempelajari literatur-literatur yang berhubungan dengan materi penyusunan aplikasi *timer* elektronik.

#### 2.1.2 Teknik Analisis Sistem

Dalam menganalisis aplikasi yang dirancang ini, penulis menggunakan UnifiedModelingLanguage (UML) untuk menggambarkan alur kerja dari aplikasi.

#### 2.1.3 Teknik Perancangan Sistem

Teknik perancangan sistem yang digunakan penulis dalam merancang aplikasi *timer* elektronik berbasis *hardware* dengan menggunakan Arduino Kit untuk mengendalikan rangkaian *relay* adalah menggunakan Microsoft Visual Basic .Net 2010 (bahasa pemrograman untuk membuat aplikasi *timer* pada komputer dengan target lingkungan kerja aplikasi menggunakan .Net Framework 4), *Integrated Development Environment* (IDE) Arduino v1.5.5-r2 (untuk menulis dan meng-*upload* kode program ke mikrokontroler pada Arduino Kit) dan Proteus 7 Professional (untuk menggambar skema rangkaian *relay* dan membuat penjaluran *Printed Circuit Board* (PCB) untuk rangkaian *relay*).

### 3.2. Landasan Teori

#### 2.2.1 Otomatisasi

Otomatisasi adalah suatu tindakan untuk mengubah perilaku dari sebuah alat, sehingga dapat melakukan fungsi yang seharusnya, namun dengan campur tangan *operator* (manusia) yang seminimal mungkin. Tujuan dari otomatisasi adalah untuk menyederhanakan masalah pengendalian yang dipandang rumit dan meringankan beban kerja manusia di dalamnya, sehingga aktivitas manusia akan menjadi lebih efektif dan efisien. [1]

#### 2.2.2 Timer

Secara umum, *timer* universal menyediakan fasilitas untuk menyalakan perangkat listrik/elektronik setelah melewati waktu-waktu tertentu, misalnya 5 menit. [2]

#### 2.2.3 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer yang dibangun pada sebuah keping (*chip*) tunggal. Jadi, hanya dengan sebuah keping IC saja dapat dibuat sebuah sistem komputer yang dapat dipergunakan untuk mengontrol alat. [3]

#### 2.2.4 Arduino

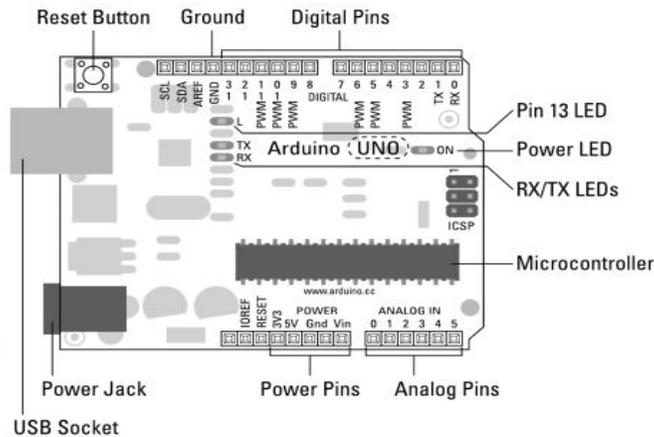
Arduino adalah *platform physical computing open source* yang berdasarkan kepada papan *input/output* (I/O) sederhana dan lingkungan pengembangan yang mengimplementasikan bahasa

*Processing* ([www.processing.org](http://www.processing.org)). Arduino dapat digunakan untuk mengembangkan proyek interaktif mandiri atau dapat dihubungkan ke perangkat lunak pada komputer (seperti Flash, Processing, VVVV, atau Max/MSP). [4]

### 2.2.5 Arduino UNO R3

Memutuskan *board* apa yang akan digunakan bisa menjadi sesuatu yang berat karena jenis *board* terus bertambah, masing-masing dengan prospek yang baru dan menarik. Namun, ada satu *board* yang dapat dianggap sebagai tulang punggung *hardware* Arduino; ini adalah yang biasanya digunakan hampir semua orang dan cocok untuk sebagian besar aplikasi. Ia adalah Arduino Uno. *Mainboard* yang terbaru sampai saat ini adalah Arduino Uno R3 (dirilis pada tahun 2011). Uno adalah bahasa Italia untuk angka satu, nama untuk versi rilis 1.0 dari *software* Arduino. R3 berkaitan dengan revisi fitur pada *board*, yang meliputi *update*, penyempurnaan, dan perbaikan. Dalam kasus ini, ini adalah revisi ketiga. [5]

Gambar berikut memperlihatkan sejumlah bagian penting dari Kit Arduino: [5]



Gambar 1 Bagian-Bagian Penting Arduino Uno

### 2.2.6 Relay

Relay adalah perangkat elektromekanis yang menggunakan elektromagnet untuk membuka atau menutup sebuah saklar. Sirkuit yang mentenagai kumparan elektromagnet sepenuhnya terpisah dari sirkuit yang meng-*on* atau *off* kan saklar pada relay, jadi dimungkinkan untuk menggunakan kumparan relay yang hanya membutuhkan beberapa *volt* untuk meng-*on* atau *off*-kan tegangan pada sirkuit. [6]

### 2.2.7 Resistor

Resistor adalah komponen elektrik yang berfungsi untuk memberikan hambatan terhadap arus listrik. Resistor memberikan hambatan agar komponen yang diberi tegangan tidak dialiri dengan arus yang besar. Ia juga dapat digunakan sebagai pembagi tegangan. [7]

### 2.2.8 Transistor

Transistor adalah komponen semikonduktor dengan tiga terminal. Sebuah *input* antara dua terminal dapat mengubah jumlah arus yang mengalir ke atau dari terminal ketiga. [8]

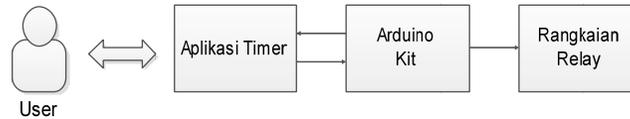
### 2.2.9 Hukum Ohm

Hukum menunjukkan bahwa arus (*I*) yang mengalir dalam suatu penghantar logam sebanding dengan beda potensial (*V*) di kedua ujungnya, asalkan suhu penghantar tetap. Besarnya kuat arus yang mengalir dalam suatu penghantar tidak hanya bergantung pada beda potensial, tetapi juga ditentukan oleh tahanan kawat penghantar (*R*). Jika dilakukan kombinasi dengan hubungan antara kuat arus dan beda potensial maka diperoleh hubungan yang dikenal sebagai hukum Ohm. [9]

## 3. HASIL DAN PEMBAHASAN

### 3.1. Analisis Prinsip Kerja Sistem

Sistem *timer* atau pewaktu yang akan dibuat penulis terdiri dari dua komponen utama yang saling berhubungan yaitu perangkat lunak (meliputi perangkat lunak pada komputer (aplikasi *timer*) dan perangkat lunak pada mikrokontroler) dan perangkat keras (terdiri dari Arduino Kit dan rangkaian *relay*).



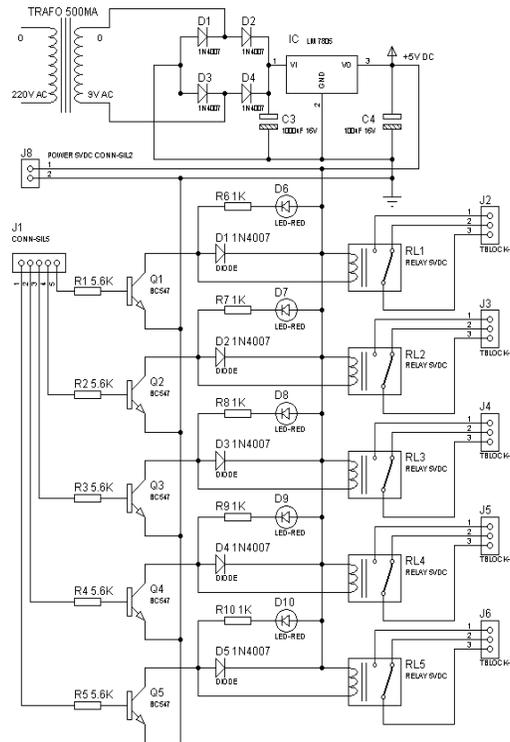
Gambar 2 Prinsip Kerja Sistem Secara Umum

Dalam pengoperasiannya, aplikasi *timer* berkomunikasi dengan mikrokontroler Arduino melalui kabel USB yang terdeteksi sebagai *communicationport* (COM port) atau *serialport* oleh komputer pada sistem operasi Windows. Pengguna dapat mengatur rentang waktu untuk menghidupkan atau mematikan rangkaian *relay* melalui antarmuka aplikasi *timer*. Pada saat rentang waktu yang ditentukan oleh pengguna tercapai, maka komputer melalui aplikasi *timer* akan mengirimkan sejumlah instruksi menuju *serialport* dan diterima oleh mikrokontroler. Hal ini akan mengakibatkan mikrokontroler memproses *input* yang diterimanya dan menghasilkan *output* logika ke rangkaian *relay*. Rangkaian *relay* ini kemudian akan menghubungkan atau memutuskan arus listrik pada beban sesuai dengan *output* logika dari mikrokontroler menurut rentang waktu yang sudah ditentukan oleh pengguna.

### 3.2. Rancangan Sistem

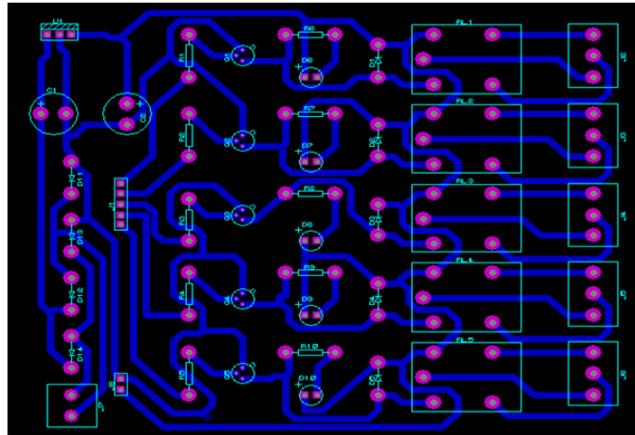
#### 3.1. Rancangan Perangkat Keras

Dalam penelitian ini, rancangan perangkat keras dibagi menjadi rancangan rangkaian *relay* dan rancangan catu daya sebagai sumber daya listrik yang memungkinkan dioperasikannya rangkaian *relay*. Keseluruhan rancangan ditunjukkan pada skema berikut ini:



Gambar 3 Skema Rangkaian Relay beserta Catu Daya

Desain skematis ini kemudian dijadikan dasar penggambaran jalur-jalur pada PCB (*Printed Circuit Board*), seperti ditunjukkan pada Gambar 4. Tampilan rangkaian *relay* dan catu daya dari atas ditunjukkan pada Gambar 5.



Gambar 4 Penjaluran PCB dan Tata Letak Komponen untuk Rangkaian Relay dan Catu Daya



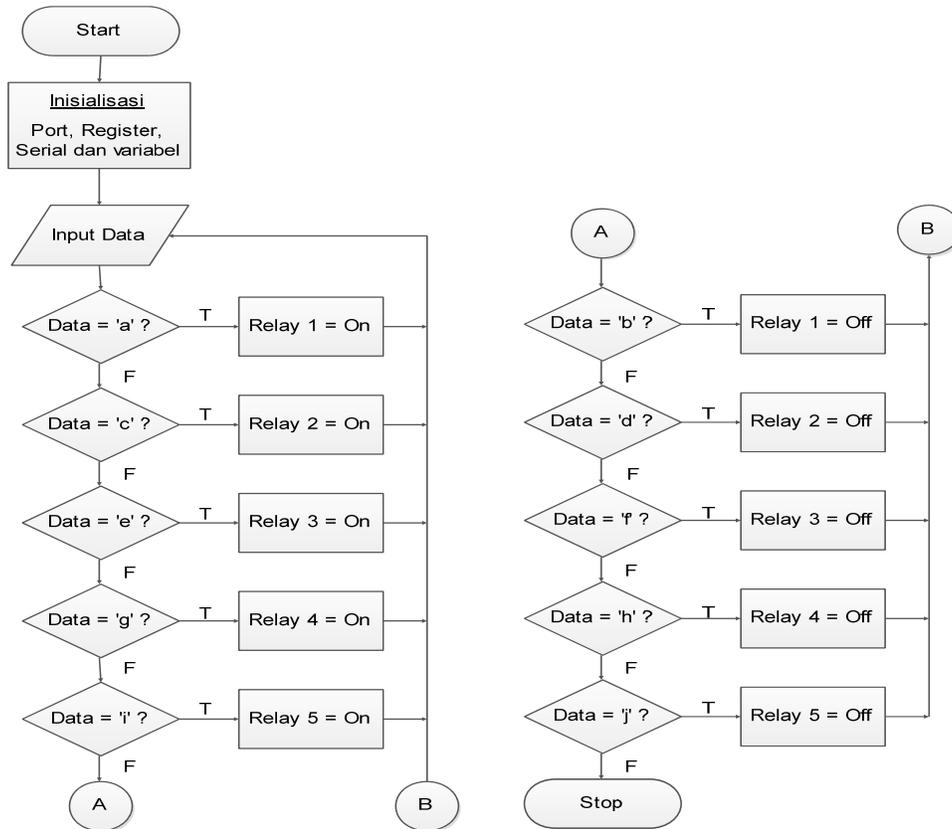
Gambar 5 Rangkaian Relay dan Catu Daya Tampak Atas

### 3.2. Rancangan Perangkat Lunak

Rancangan perangkat lunak dibagi menjadi dua subbagian, sebagai berikut:

#### 3.2.2.1 Rancangan Perangkat Lunak untuk Mikrokontroler

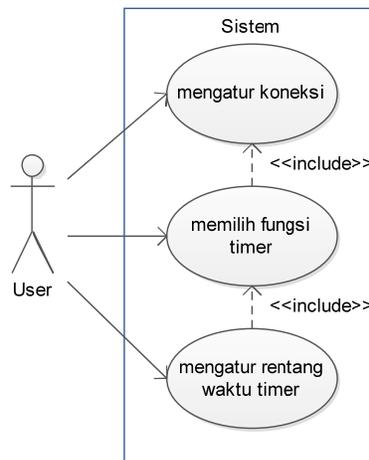
Agar Arduino kit dapat berkomunikasi dan memproses perintah yang dikirimkan kepadanya, maka mikrokontroler pada Arduino kit harus diprogram terlebih dahulu. Pemrograman mikrokontroler pada Arduino kit dapat dilakukan melalui *IntegratedDevelopmentEnvironment (IDE)* yang disediakan oleh Arduino (<http://arduino.cc/en/main/software>) secara gratis untuk sistem operasi Windows, Macintosh dan Linux. Editor IDE Arduino yang digunakan penulis pada penelitian ini secara *default* terbagi menjadi dua bagian, yaitu bagian *voidsetup* (untuk penulisan kode program yang dijalankan hanya satu kali) dan *voidloop* (untuk menyetikkan kode program yang dijalankan berulang-ulang). Penulisan kode program dimulai dengan mendeklarasikan nomor pin yang akan dipakai pada bagian teratas editor. Penulis menggunakan lima digital pin yaitu pin dengan nomor 3, 4, 5, 6 dan 7. Pada bagian *voidsetup*, pin yang sudah dideklarasikan tadi diset sebagai *output*. Kemudian pada bagian *voidloop*, diketikkan kode program yang mengatur nilai *output* digital pin yaitu *high* atau *low* sesuai dengan perintah yang dikirimkan aplikasi *timer* melalui *COM port*. Setelah kode program selesai diketik, pengguna harus meng-*upload* kode program ke mikrokontroler dengan cara mengklik tombol *upload*. Logika dasar yang digunakan dalam merancang kode program untuk mikrokontroler Arduino ini ditunjukkan pada *flowchart* sebagai berikut:



Gambar 6 Flowchart Cara Kerja Mikrokontroler pada Arduino

### 3.2.2.2 Rancangan Perangkat Lunak pada Komputer

Rancangan aplikasi *timer* (perangkat lunak pada komputer) dalam penelitian ini dapat dimodelkan secara umum menggunakan *Unified Modeling Language* (UML) berupa *Use Case Diagram*. *Use Case Diagram* ditunjukkan pada Gambar 7, dan secara keseluruhan menunjukkan interaksi *user* dengan sistem. Saat aplikasi dijalankan, *user* harus memilih COM *port* yang sesuai agar dapat terhubung dengan Arduino Kit. Pada saat aplikasi dan Arduino Kit sudah terhubung, *user* dapat memilih fungsi *timer*, yaitu untuk menghidupkan atau mematikan rangkaian *relay*. Setelah fungsi *timer* dipilih, *user* dapat mengatur rentang waktu pengaktifan *timer*.



Gambar 7 Use Case Diagram

3.3. Tampilan Interface Utama Aplikasi

Gambar berikut menunjukkan tampilan antarmuka utama aplikasi:



Gambar 8 Tampilan Form Utama Dan Bagian Menu Utama

Untuk menjalankan aplikasi pengguna dapat membuka pintasan aplikasi Timer Arduino v1.0 yang terletak di *desktop* sistem operasi. Setelah dibuka, maka *splashscreen* akan ditampilkan sekilas, yang menandakan bahwa aplikasi telah dijalankan dan aktif di memori sistem operasi Windows. Setelah *splashscreen* menghilang, maka *form* utama akan ditampilkan. *Form* utama secara garis besar terdiri dari tiga menu utama, yaitu menu pengaturan koneksi, menu pengaturan *timer* dan menu informasi *timer*.

3.4. Pengujian Aplikasi

Penulis menguji keakuratan pewaktuan oleh aplikasi dengan membandingkan waktu yang diatur pada aplikasi *timer* dengan waktu yang diukur dengan menggunakan *stopwatch*. Dalam melakukan pengujian, penulis melakukan pengujian dengan perangkat sebagai berikut:

- a. Notebook Intel Celeron Dual Core, RAM 4GB, *hardisk* 500GB, *port* USB 2.0.
- b. Arduino UNO R3 yang sudah diprogram dan sudah dirangkai ke rangkaian *relay*.
- c. *Stopwatch* sebagai pembanding waktu.
- d. Tahanan beban berupa lima buah bola lampu 220V AC, dengan daya masing-masing 5W yang dihubungkan ke rangkaian *relay*.

Pengujian dilakukan terhadap semua *timer* dengan fungsi dan pengaturan *timer* sebagai berikut:

Tabel 1 Fungsi Dan Pengaturan Timer Yang Dipilih Untuk Pengujian

No	Fungsi Timer	Pengaturan Waktu
1	Matikan relay	Waktu spesifik
2	Matikan relay	Dari sekarang
3	Hidupkan relay	Waktu spesifik
4	Hidupkan relay	Dari sekarang

3.4.1 Matikan Relay dan Waktu Spesifik

Pada pengujian dengan pilihan Matikan Relay dan Waktu Spesifik, penulis menggunakan variabel sebagai berikut:

3.4.1.1 Timer, adalah *tab* Timer yang diuji.

- 3.4.1.2 Jam Sistem, adalah jam berjalan pada sistem yang ditunjukkan pada label sebelah kiri bawah pada aplikasi Timer Arduino v1.0 dengan format hh:mm:ss.
- 3.4.1.3 Waktu Spesifik Timer, adalah waktu yang diatur pada setiap timer dengan format dd/mm/yyyy, hh:mm:ss.
- 3.4.1.4 Selisih waktu, merupakan selisih waktu antara jam sistem dengan waktu spesifik timer yang diset selang 5 menit.
- 3.4.1.5 Stopwatch, adalah selisih nilai yang ditampilkan pada stopwatch sejak tombol Aktifkan diklik sampai dengan messageboxtimer aktif berhasil ditampilkan.

Hasil pengujian menunjukkan bahwa tahanan beban ( $R_L$ ) berupa lampu dapat menyala selama 5 menit sejak tombol Aktifkan pada aplikasi diklik dan mati setelah rentang waktu yang ditentukan habis. Selisih yang terjadi antara rentang waktu spesifik *timer* dan *stopwatch* dapat diabaikan karena sangat kecil.

Tabel 2 Matikan Relay Pada Waktu Spesifik

Timer	Jam Sistem hh:mm:ss	Waktu Spesifik Timer dd/mm/yyyy, hh:mm:ss	Selisih Waktu	Stopwatch mm:ss:1/10s
Timer 1	09:30:00	7/4/2014, 09:35:00	5 menit	05:00:8
Timer 2	09:38:00	7/4/2014, 09:43:00	5 menit	04:59:3
Timer 3	09:45:00	7/4/2014, 09:50:00	5 menit	05:00:3
Timer 4	09:55:00	7/4/2014, 10:00:00	5 menit	04:59:5
Timer 5	10:03:00	7/4/2014, 10:08:00	5 menit	05:00:2
Semua Timer	10:10:00	7/4/2014, 10:15:00	5 menit	05:00:4

3.4.2 Matikan Relay dan Dari Sekarang

Pada pengujian dengan pilihan Matikan Relay dan Dari Sekarang, penulis menggunakan variabel sebagai berikut:

- 3.4.2.1 Timer, adalah *tab* Timer yang diuji.
- 3.4.2.2 Jam Sistem, adalah jam berjalan pada sistem yang ditunjukkan pada label sebelah kiri bawah pada aplikasi Timer Arduino v1.0 dengan format hh:mm:ss.
- 3.4.2.3 Waktu Timer, adalah waktu yang diatur lima menit dari sekarang pada setiap timer dengan format hh:mm.
- 3.4.2.4 Jam Aktif, merupakan jumlah dari Jam Sistem ditambah Waktu Timer.
- 3.4.2.5 Stopwatch, adalah selisih nilai yang ditampilkan pada stopwatch sejak tombol Aktifkan diklik sampai dengan messageboxtimer aktif berhasil ditampilkan.

Hasil pengujian menunjukkan bahwa tahanan beban ( $R_L$ ) berupa lampu yang menyala sejak tombol Aktifkan diklik dapat dimatikan setelah rentang waktu lima menit. Selisih waktu antara waktu *timer* dengan *stopwatch* sangat kecil, kurang lebih satu detik sehingga dapat diabaikan.

Tabel 3 Matikan Relay Dari Sekarang

Timer	Jam Sistem hh:mm:ss	Waktu Timer hh:mm	Jam Aktif hh:mm	Stopwatch mm:ss:1/10s
Timer 1	10:30:00	00:05	10:35	04:59:8
Timer 2	10:40:00	00:05	10:45	05:00:6
Timer 3	10:50:00	00:05	10:55	04:59:5
Timer 4	11:00:00	00:05	11:05	05:00:5
Timer 5	11:10:00	00:05	11:15	05:00:6
Semua Timer	11:25:00	00:05	11:30	04:59:5

3.4.3 Hidupkan Relay dan Waktu Spesifik

Pada pengujian dengan pilihan Hidupkan Relay dan Waktu Spesifik, penulis menggunakan variabel sebagai berikut:

- 3.4.3.1 Timer, adalah *tab* Timer yang diuji.
- 3.4.3.2 Jam Sistem, adalah jam berjalan pada sistem yang ditunjukkan pada label sebelah kiri bawah pada aplikasi Timer Arduino v1.0 dengan format hh:mm:ss.
- 3.4.3.3 Waktu Spesifik Timer, adalah waktu yang diatur pada setiap *timer* dengan format dd/mm/yyyy, hh:mm:ss.
- 3.4.3.4 Selisih waktu, merupakan selisih waktu antara jam sistem dengan waktu spesifik *timer* yang diset selang 5 menit.

3.4.3.5 Stopwatch, adalah selisih nilai yang ditampilkan pada *stopwatch* sejak tombol Aktifkan diklik sampai dengan *messageboxtimer* aktif berhasil ditampilkan.

Hasil pengujian menunjukkan bahwa tahanan beban ( $R_L$ ) berupa lampu dapat menyala setelah rentang waktu yang ditentukan pengguna habis. Selisih yang terjadi antara rentang waktu spesifik *timer* dan *stopwatch* dapat diabaikan karena sangat kecil.

Tabel 4 Hidupkan Relay Pada Waktu Spesifik

Timer	Jam Sistem hh:mm:ss	Waktu Spesifik Timer (dd/mm/yyyy, hh:mm:ss)	Selisih Waktu	Stopwatch mm:ss:1/10s
Timer 1	13:30:00	7/4/2014, 13:35:00	5 menit	05:00:3
Timer 2	13:38:00	7/4/2014, 13:43:00	5 menit	05:00:5
Timer 3	14:00:00	7/4/2014, 14:05:00	5 menit	04:59:5
Timer 4	14:10:00	7/4/2014, 14:15:00	5 menit	04:59:5
Timer 5	14:18:00	7/4/2014, 14:23:00	5 menit	05:00:2
Semua Timer	14:30:00	7/4/2014, 14:35:00	5 menit	05:00:4

3.4.4 Hidupkan Relay Dan Dari Sekarang

Pada pengujian dengan pilihan Hidupkan Relay dan Dari Sekarang, penulis menggunakan variabel sebagai berikut:

3.4.4.1 Timer, adalah *tab* Timer yang diuji.

3.4.4.2 Jam Sistem, adalah jam berjalan pada sistem yang ditunjukkan pada label sebelah kiri bawah pada aplikasi Timer Arduino v1.0 dengan format hh:mm:ss.

3.4.4.3 Waktu Timer, adalah waktu yang diatur lima menit dari sekarang pada setiap *timer* dengan format hh:mm.

3.4.4.4 Jam Aktif, merupakan jumlah dari Jam Sistem ditambah Waktu Timer.

3.4.4.5 Stopwatch, adalah selisih nilai yang ditampilkan pada *stopwatch* sejak tombol Aktifkan diklik sampai dengan *messageboxtimer* aktif berhasil ditampilkan.

Hasil pengujian menunjukkan bahwa tahanan beban ( $R_L$ ) berupa lampu yang menyala sejak tombol Aktifkan diklik dapat dimatikan setelah rentang waktu lima menit. Selisih waktu antara waktu *timer* dengan stopwatch sangat kecil, kurang lebih satu detik sehingga dapat diabaikan.

Tabel 5 Hidupkan Relay Dari Sekarang

Timer	Jam Sistem hh:mm:ss	Waktu Timer hh:mm	Jam Aktif hh:mm	Stopwatch mm:ss:1/10s
Timer 1	15:00:00	00:05	15:05	04:59:8
Timer 2	15:10:00	00:05	15:15	05:00:6
Timer 3	15:20:00	00:05	15:25	04:59:5
Timer 4	15:30:00	00:05	15:35	05:00:5
Timer 5	15:40:00	00:05	15:45	05:00:6
Semua Timer	15:50:00	00:05	15:55	04:59:5



Gambar 9 Ilustrasi Kondisi Sewaktu Pengujian Semua Relay Off



Gambar 10 Ilustrasi Kondisi Sewaktu Pengujian Semua Relay On

Berdasarkan data hasil pengujian, penulis menyimpulkan bahwa aplikasi *timer* yang penulis rancang telah dapat berfungsi baik dan sesuai dengan harapan penulis. Selisih waktu yang terjadi antara pengaturan waktu pada *timer* dan waktu pada *stopwatch* dapat dikarenakan pada saat pengujian terjadi selisih beberapa milidetik antara penekanan tombol Aktifkan pada *tabtimer* dengan penekanan tombol pada *stopwatch*. Selisih yang terjadi memiliki nilai yang sangat kecil dan dapat diabaikan, karena perancangan aplikasi *timer* oleh penulis dibatasi memiliki keakuratan sampai menit saja.

#### 4. KESIMPULAN

- a. Aplikasi *timer* dan rangkaian *relay* yang dirancang oleh penulis cocok digunakan untuk penjadwalan proses kerja yang berulang dan tidak memerlukan akurasi waktu yang tinggi.
- b. Berdasarkan *datasheet* relay Songle (<http://www.songle.com/en/pdf/20084141716341001.pdf>), batas daya maksimum yang dapat dibebankan pada *relay* adalah 1.680 Watt pada tegangan 240V AC dan 280 Watt pada tegangan 28V DC untuk setiap unit *relay*. Menghubungkan beban yang memiliki daya di atas kemampuan *relay* dapat merusak komponen *relay* dan mengakibatkan kegagalan proses kerja keseluruhan sistem.
- c. Pengaturan fungsi *timer* dan rentang waktu *timer* hanya berlaku selama aplikasi dijalankan, putusnya sumber arus listrik pada rangkaian *relay* atau matinya komputer yang menjalankan aplikasi akan menyebabkan kegagalan kerja sistem. Pengguna harus melakukan pengaturan ulang terhadap fungsi *timer* dan rentang waktu pengaktifan *timer* jika hal tersebut terjadi.

#### 5. SARAN

Untuk mengembangkan sistem ini lebih lanjut, ada beberapa hal yang disarankan:

- a. Aplikasi yang ditulis penulis tidak menggunakan *database* untuk menyimpan semua pengaturan yang dilakukan oleh pengguna. Sehingga jika terjadi kegagalan pada sistem kelistrikan yang menyebabkan komputer padam, maka pengguna harus mengatur ulang aplikasi. Dengan menggunakan *database* diharapkan hal ini dapat ditanggulangi.
- b. Ketepatan waktu pada *timer* masih membutuhkan pengujian lebih lanjut, karena aplikasi ditulis dengan komponen standar *timer* pada bahasa pemrograman Visual Basic .Net.
- c. Kode program aplikasi masih dapat dilakukan penyederhanaan secara struktur dan logika.
- d. *Input* logika pada rangkaian *relay* terhubung secara langsung ke mikrokontroler pada Arduino kit. Hal ini dapat menyebabkan kerusakan pada mikrokontroler jika terjadi kerusakan pada rangkaian *relay*. Untuk mengatasi hal ini, penulis menyarankan untuk menggunakan komponen *optocoupler* pada *input* logika. Sehingga secara kelistrikan, tidak ada kontak langsung antara mikrokontroler dengan rangkaian *relay*.

#### DAFTAR PUSTAKA

- [1] Wahana Komputer (2006). *Teknik Antarmuka Mikrokontroler dengan Komputer Berbasis Delphi*. Salemba Infotek. Jakarta.
- [2] EFY Enterprises Pvt Ltd, (2007), *Electronics Projects Vol. 24*, Ramesh Chopra, New Delhi.

- [3] Malik, Moh. Ibnu dan Mohammad Unggul Juwana. (2009). *Aneka Proyek Mikrokontroler PIC16F84/A*. PT Elex Media Komputindo. Jakarta.
- [4] Banzi, Massimo. (2011). *Getting Started With Arduino by Massimo Banzi. 2nd Edition*. O'Reilly. North Sebastopol.
- [5] Nussey, John. (2013). *Arduino® For Dummies®*. John Wiley & Sons. England.
- [6] Lowe, Doug. (2012). *Electronics All-in-One For Dummies®*. John Wiley & Sons. New Jersey.
- [7] Budiharto, Widodo. (2008). *10 Proyek Robot Spektakuler. PT Elex Media Komputindo*. Jakarta.
- [8] Sinclair, Ian. (2011). *Electronics Simplified*. Elsevier. United Kingdom.
- [9] Umar, Efrizon. (2008). *Buku Pintar Fisika*. Media Pusindo. Jakarta.