SKOPIN (STOP KONTAK PINTAR) PENGENDALI ARUS LISTRIK MENGGUNAKAN TIMER PADA STOP KONTAK BERBASIS ARDUINO

SKRIPSI

Disusun oleh:
ALITINIA PRASTIANTARI
3145136208

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA
JAKARTA

2017

HALAMAN PERSEMBAHAN

Untuk Ibu, Bapak, kakak-kakakku, keponakan-keponakanku,

dan sahabat-sahabatku tercinta.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur penulis panjatkan ke hadirat Allah SWT karena hanya dengan rahmat dan hidayah-Nya, Tugas Akhir ini dapat terselesaikan tanpa halangan berarti. Keberhasilan dalam menyusun laporan Tugas Akhir ini tidak lepas dari bantuan berbagai pihak yang mana dengan tulus dan ikhlas memberikan masukan guna sempurnanya Tugas Akhir ini. Oleh karena itu dalam kesempatan ini, dengan kerendahan hati penulis mengucapkan terima kasih kepada:

1. Bapak Drs. Mulyono, M.Kom., selaku Ketua Program Studi Ilmu Komputer Universitas Negeri Jakarta dan dosen pembimbing kedua yang telah memberikan banyak bantuan, bimbingan, serta arahan dalam Tugas Akhir dan kegiatan-kegiatan yang lain,

2. Ibu Ir. Fariani Hermin M.T., selaku dosen pembimbing pertama yang telah memberikan banyak bantuan, bimbingan, serta arahan dalam Tugas Akhir ini,

3. Bapak Ibnu Hadi, M.Si. selaku dosen pembimbing akademik penulis,

4. Bapak Muhammad Eka Suryana, M.Kom, serta Bapak Agus Agung Permana, S.Si selaku dosen sekaligus pembina DEFAULT yang telah memberikan banyak ilmu selama penulis tergabung sebagai anggota,

5. Seluruh Dosen di Program Studi Ilmu Komputer dan FMIPA UNJ, yang tidak bisa disebutkan satu per satu, atas ilmu dan bimbingannya selama penulis berkuliah di Ilmu Komputer FMIPA UNJ,

6. Ibu Tri Astuti dan Bapak Suprapto yang selama ini telah sabar membimbing,

mengarahkan, dan mendoakan penulis tanpa kenal lelah sehingga penulis dapat menyelesaikan Tugas Akhir ini,

7. Mas Fara, Mba Rita, Mba Ita, Mas Kus, Mba Anes, Mas Agung, Naya, Zee, Fay, Jojo, Zeta, kakak-kakak serta keponakan-keponakan yang senantiasa mendoakan dan memberikan dukungan kepada penulis,

8. M. Reyhan Fahlevi yang selalu membimbing, menyemangati, dan mendoakan penulis dalam masa pembuatan Tugas Akhir ini,

9. Teman-teman Ilmu Komputer 2013 khususnya TAKADA atas dorongan, semangat serta penghiburan yang senantiasa diberikan kepada penulis dalam keadaan suka maupun duka,

10. Dan seluruh kerabat yang tidak dapat disebutkan satu per satu oleh penulis atas dukungan serta doa yang diberikan kepada penulis.

Penulis menyadari bahwa penyusunan Proposal Tugas Akhir ini jauh dari sempurna. Kritik dan saran dapat ditujukan langsung pada e-mail *alitiniahasan@yahoo.com*. Akhir kata penulis mohon maaf apabila ada kekeliruan di dalam penulisan Tugas Akhir ini. Terimakasih.

Wassalamu'alaikum Wr. Wb.

Jakarta, 3 Agustus 2017

Penulis

DAFTAR ISI

HALAMAN PERSEMBAHAN **ii**

KATA PENGANTAR **iii**

DAFTAR ISI **vii**

DAFTAR TABEL **viii**

DAFTAR GAMBAR **x**

Abstrak **xi**

Abstract **xii**

I PENDAHULUAN **1**

A. Latar Belakang Masalah 1

B. Identifikasi Masalah 3

C. Batasan Masalah 4

D. Perumusan Masalah 4

E. Tujuan Penelitian 4

F. Manfaat Penelitian 5

II KAJIAN PUSTAKA **6**

A. Arduino 6

1. Arduino *Software* 6

2. Arduino *Hardware* 9

3. *Library* Arduino 13

B. Sensor Air 15

C. Relay 16
D. LCD (Liquid Crystal Display) 18
1. Pengendali / Kontroler LCD (Liquid Cristal Display) 19
E. FSM (*Finite State Machine*) 20
1. Diagram Keadaan 21

III IMPLEMENTASI PROGRAM **29**
A. Tempat dan Waktu Penelitian 29
B. Metode Penelitian 29
1. Alat dan Bahan 29
2. Prosedur Penelitian 31
C. Teknik Pengumpulan dan Analisis Data 35

IV PERCOBAAN DAN HASIL PERCOBAAN **37**
A. Tahapan Uji Coba 37
B. Hasil Percobaan 40
1. Hasil Percobaan Keakuratan Waktu & fungsi Timer 40
2. Hasil Percobaan Batas Air pada Sensor Air 44

V KESIMPULAN DAN SARAN **57**
A. Kesimpulan 57
B. Saran 58

DAFTAR PUSTAKA **60**

LAMPIRAN **61**

A **61**

B **64**

C *Source Code* 69

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino UNO R3 9

Tabel 2.2 Perbedaan *Mealy dan Moore Machine* 25

Tabel 2.3 Tabel Keadaan 26

Tabel 2.4 Penejelasan diagram FSM. 28

Tabel 4.1 Data pengetesan keakuratan *timer* SKOPIN terminal A 39

Tabel 4.2 Data pengetesan keakuratan *timer* SKOPIN terminal B 39

Tabel 4.3 Hasil percobaan pada SKOPIN teriminal A 41

Tabel 4.4 Hasil percobaan pada SKOPIN terminal B 42

Tabel 1.1 Hasil percobaan probabilitas selisih *timer* SKOPIN pada waktu 15 detik dengan *stopwatch* pada *smartphone* 62

DAFTAR GAMBAR

Gambar 2.1 Tampilan ikon aplikasi *software* Arduino IDE 7

Gambar 2.2 Tampilan aplikasi *software* Arduino IDE 7

Gambar 2.3 Sensor air 16

Gambar 2.4 Relay yang akan digunakan yaitu 2 Relay Module. 17

Gambar 2.5 LCD yang akan digunakan yaitu Modul LCD Keypad. 18

Gambar 2.6 Contoh diagram *state*. 22

Gambar 2.7 *Moore State Machine*. 24

Gambar 2.8 *Mealy State Machine*. 24

Gambar 2.9 Diagram rancang bangun FSM SKOPIN. 26

Gambar 3.1 Alat dan bahan yang digunakan untuk pembuatan skopin . . . 29

Gambar 3.2 Desain Rancangan Alat. 32

Gambar 3.3 Diagram Rancangan Alat. 33

Gambar 3.4 Diagram alur SKOPIN. 34

Gambar 3.5 Diagram Tahapan Pelaksanaan. 35

Gambar 4.1 Tampilan SKOPIN dari atas 38

Gambar 4.2 *Console* Arduino IDE pada saat sensor air kering 45

Gambar 4.3 Tetesan banyak air pada sensor 46

Gambar 4.4 *Console* Arduino IDE pada saat sensor air terkena tetesan air yang banyak 46

Gambar 4.5 Sensor dimasukkan 1 mm ke dalam air 47

Gambar 4.6 *Console* Arduino IDE pada saat sensor air dimasukkan 1mm ke dalam air 48

Gambar 4.7 Tampilan layar LCD SKOPIN pada saat terdeteksi air 49

Gambar 4.8 Tetesan sedikit air pada sensor 49

Gambar 4.9 *Console* Arduino IDE pada saat sensor air terkena percikan air 50
Gambar 4.10 Tampilan *home* SKOPIN 50
Gambar 4.11 Sensor dimasukkan 0,5 cm ke dalam air 51
Gambar 4.12 *Console* Arduino IDE pada saat sensor air 0,5 cm ke dalam air 51
Gambar 4.13 Sensor dimasukkan 1 cm ke dalam air 52
Gambar 4.14 *Console* Arduino IDE pada saat sensor air 1 cm ke dalam air . 52
Gambar 4.15 Sensor dimasukkan 2 cm ke dalam air 53
Gambar 4.16 *Console* Arduino IDE pada saat sensor air 2 cm ke dalam air . 53
Gambar 4.17 Sensor dimasukkan 3 cm ke dalam air 54
Gambar 4.18 *Console* Arduino IDE pada saat sensor air 3 cm ke dalam air . 54
Gambar 4.19 Sensor dimasukkan 3,8 cm ke dalam air 55
Gambar 4.20 *Console* Arduino IDE pada saat sensor air 3,8 cm ke dalam air 55
Gambar 4.21 Tampilan layar LCD SKOPIN pada saat terdeteksi air 56

Gambar 1.1 Rumus simpangan baku untuk data tunggal 62

Gambar 2.1 SKOPIN tampak atas 64
Gambar 2.2 SKOPIN tampak depan 64
Gambar 2.3 SKOPIN tampak samping kiri 65
Gambar 2.4 SKOPIN tampak samping kanan 65
Gambar 2.5 SKOPIN tampak belakang 66
Gambar 2.6 SKOPIN bagian dalam 67
Gambar 2.7 tampilan utama SKOPIN 68

Abstrak

Gadget khususnya *smartphone* merupakan salah satu kebutuhan utama masyarakat yang memiliki banyak manfaat, salah satunya untuk memudahkan pekerjaan. Baterai ialah komponen utama pada perangkat. Hal yang sering dilupakan oleh masyarakat ialah membiarkan perangkat mereka diisi daya (di-*charge*) terlalu lama sehingga baterai pada perangkat tersebut mengalami kelebihan daya (*overcharge*) yang dapat merusak baterai yang memungkinkan untuk mengurangi kinerja dari perangkat tersebut. Dalam skripsi ini, memanfaatkan mikrokontroler sebagai sistem pengendali arus listrik agar pengguna tidak perlu khawatir terjadi *overcharging* pada perangkat. Sistem pengendali arus listrik tersebut adalah SKOPIN (Stop Kontak Pintar). Mikrokontroler Arduino UNO R3 akan diprogram didalamnya dengan menerapkan teori FSM (Finite State Machine), untuk menghubungkan tombol pengatur waktu dan relay yang akan berfungsi untuk mengatur waktu yang digunakan untuk mengaliri listrik pada setiap perangkat. Sebagai tambahan, terdapat pula fitur pendeteksi air pada SKOPIN yang bertujuan untuk melindungi SKOPIN dan perangkat dari kerusakan akibat terkena air baik karena keteledoran manusia (tersiram air) atau bencana alam (banjir), sehingga sensor dapat langsung mengirim sinyal analog ke Arduino sehingga Arduino dapat mendeteksi air dan memutuskan arus listrik sebelum terjadi korsleting.

Kata kunci : SKOPIN, *Overcharging*, Arduino, FSM (*Finite State Machine*).

Abstract

Gadget in particular smartphone is one of the main needs of people who have many benefits, one of them to facilitate the work. Battery is the main component of the device. The thing that people often forget is to leave their devices charged for too long so that the batteries on the device are overcharge that can damage the battery that makes it possible to reduce the performance of the device. In this thesis, utilizing microcontroller as an electric current controller system so that users do not have to worry about overcharging on the device. The electrical current control system is SKOPIN (Stop Kontak Pintar). The Arduino UNO R3 microcontroller will be programmed therein by applying the Finite State Machine (FSM) theory, to connect the timer and relay buttons that will serve to adjust the time used to power each device. In addition, there is also a water detection feature in SKOPIN aimed at protecting SKOPIN and the device from damage caused by water exposure either due to human neglect (water hazard) or natural disaster (flood), so that the sensor send analog signal to Arduino and Arduino can directly detect water and disconnect the electricity before the short circuit.

Keywords : SKOPIN, *Overcharging*, Arduino, FSM (*Finite State Machine*).

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Kemajuan teknologi yang terjadi di Dunia khususnya Indonesia tidak lepas dari perkembangan yang terjadi pada alat-alat elektronika. Peralatan elektronika digemari oleh masyarakat karena dapat membantu pekerjaan manusia dan membuat berbagai hal menjadi lebih efektif dan efisien. Pada penggunaannya, peralatan elektronika dalam kehidupan sehari-hari selalu menggunakan batasan waktu tertentu. Dalam pengaturan waktu tersebut, ada yang diatur oleh pengguna sesuai dengan kebutuhan, ada juga yang sudah tersedia alat pengatur waktu (*timer*) oleh pembuat alat tersebut (standar pabrik). Beberapa contoh peralatan elektronika yang dapat dikendalikan oleh penggunanya adalah televisi, *microwave*, penanak nasi, *air conditioner* dan masih banyak lainnya. Tetapi banyak juga peralatan yang tidak dapat diatur waktu saat *on/off*-nya, sehingga pengguna harus mengingat kapan waktu yang tepat untuk mematikan peralatan tersebut setelah beberapa waktu seperti mengisi daya pada baterai *handphone*, laptop, tablet, dan masih banyak lagi.

Menurut Kementerian Komunikasi dan Informatika Republik Indonesia, Lembaga riset digital marketing Emarketer memperkirakan pada 2018 jumlah pengguna aktif *smartphone* di Indonesia lebih dari 100 juta orang. Dengan jumlah sebesar itu, Indonesia akan menjadi negara dengan pengguna aktif *smartphone* terbesar keempat di dunia setelah Cina, India, dan Amerika [11]. Sebagian besar orang tidak mengawasi *gadget* khususnya *smartphone* yang sedang di-*charge* melainkan ditinggal untuk melakukan aktifitas lain ataupun ditinggal tidur semalaman yang akan membuat perangkat tersebut mengalami *overcharge*. *Overcharging* adalah kondisi dimana pro-

ses pengisian daya tetap dilanjutkan meskipun baterai sudah terisi penuh. Ketika pengisi daya baterai dihubungkan secara paralel dengan baterai, kapasitansi baterai akan mencegah kontribusi pengisi baterai dari kenaikan seketika. Namun, apabila perangkat dibiarkan terlalu lama terisi daya, maka baterai akan menerima daya diluar kapasitasnya yang dapat mengakibatkan kerusakan [5]. Menurut penelitian, baterai *lithium ion* yang biasa digunakan pada *smartphone* tidak boleh dibiarkan terhubung dengan listrik terlalu lama ketika dayanya sudah terisi 100%. *Overcharging* dapat memproduksi karbon dioksida, etilena dan gas lainnya sehingga meningkatkan suhu dan tekanan internal yang menyebabkan baterai tersebut tidak dapat menyimpan daya dengan baik, menggembung, menjadi cepat rusak, bahkan dalam kasus terburuk dapat terjadi ledakan sel [6].

Selain *Overcharging*, kebiasaan masyarakat di Indonesia adalah meninggalkan *charger* menetap pada stop kontak ketika selesai mengisi daya dan sudah mencabut perangkat dari *charger* tersebut. Penelitian yang dilakukan Lawrence Berkeley National Library pada tahun 2012 menyebutkan bahwa rata-rata saat *charger* mengisi daya ke ponsel aliran listriknya 3,68 watt. Ketika daya ponsel sudah 100% maka dayanya turun ke 2,24 watt. Saat *charger* terhubung ke listrik tanpa ponsel, listrik akan menyerap daya 0,26 watt [7]. Penelitian ini membuktikan bahwa, listrik akan tetap menyerap daya pada saat *charger* menempel pada stop kontak, sehingga tidak menghemat penggunaan listrik. Selain itu, apabila *charger* berada pada stop kontak terlalu lama dan terjadi penumpukan steker pada stop kontak, hal tersebut dapat menyebabkan terjadinya kerusakan pada kabel-kabel dan komponen penghantar listrik karena menerima arus yang terlalu besar. Selain itu, pemakaian listrik berlebihan akan berdampak pada kurangnya pasokan listrik serta dapat menyebabkan pemanasan global [12].

Oleh karena itu, untuk menanggulangi permasalahan yang telah dipaparkan

sebelumnya, penulis memiliki gagasan untuk membuat SKOPIN (STOP KONTAK PINTAR). SKOPIN merupakan sebuah inovasi pengendali arus listrik yang menggunakan alat pengatur waktu *(timer)* pada stop kontak berbasis Arduino. Stop kontak tersebut akan berfungsi sebagai pengendali listrik yang akan diatur menggunakan *timer* otomatis yang dihubungkan dengan LCD, sehingga listrik akan terputus sesuai dengan waktu yang telah ditentukan. SKOPIN bertujuan untuk mengatur pemakaian listrik secukupnya agar dapat mencegah terjadinya kerusakan pada baterai dan mengurangi pemakaian listrik yang berlebihan pada perangkat yang diakibatkan oleh terlalu lama mencolok steker pada stop kontak. Arduino akan digunakan sebagai "otak" yang akan diberikan program didalamnya dengan menerapkan teori FSM (Finite State Machine), untuk menghubungkan tombol pengatur waktu dan relay yang akan berfungsi untuk mengatur waktu yang digunakan untuk mengaliri listrik pada setiap perangkat. Sebagai fitur tambahan, penulis menambahkan sensor air pada SKOPIN yang bertujuan untuk melindungi SKOPIN dan perangkat dari kerusakan akibat terkena air baik karena keteledoran manusia (tersiram air) atau bencana alam (banjir), sensor dapat langsung mengirim sinyal analog ke Arduino sehingga Arduino dapat mendeteksi air dan memutuskan arus listrik sebelum terjadi korsleting.

B. Identifikasi Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, masalah yang diidentifikasi sebagai berikut:

1. Banyak terjadi kerusakan pada baterai perangkat elektronik akibat terlalu lama mengisi daya (*overcharging*).
2. Pemakaian listrik yang berlebihan akan menyebabkan biaya listrik menjadi sedikit lebih mahal dan tidak menghemat penggunaan listrik.

C. Batasan Masalah

1. Pada *interface* dan *timer*-nya, SKOPIN menerapkan metode FSM (*Finite State Machine*).
2. SKOPIN menerapkan stop kontak dengan kapasitas 2 slot terminal.
3. SKOPIN dapat diatur dalam waktu 15 menit dan kelipatannya sampai dengan 4 jam.
4. Pada penerapannya, SKOPIN baik digunakan untuk peralatan elektronik yang memiliki baterai. Tidak dianjurkan untuk pemakaian peralatan elektronik yang membutuhkan daya besar setiap kali memulai aliran listrik.

D. Perumusan Masalah

1. Bagaimana cara mencegah kerusakan baterai pada perangkat akibat *overcharging*?
2. Apa upaya yang dilakukan untuk menghemat pemakaian listrik?
3. Bagaimana menerapkan metode FSM pada pembuatan SKOPIN?

E. Tujuan Penelitian

1. Mencegah kerusakan baterai pada perangkat akibat *overcharging*
2. Menghemat pemakaian dan biaya listrik
3. Menerapkan FSM (*Finite State Machine*) pada *interface* menu alat dan *timer*-nya.

F. Manfaat Penelitian

Bagi Penulis:

1. Mengetahui cara mengembangkan Arduino yang dapat diintegerasikan dengan peralatan sehari-hari, yaitu dengan membuat stop kontak yang memiliki timer sehingga arus listrik dapat terputus sesuai dengan waktu yang telah ditentukan.

Bagi Program Studi Ilmu Komputer:

1. Memberikan informasi pada mahasiswa mengenai salah satu penerapan mikro-kontroler Arduino yang dapat diterapkan sehari-hari.
2. Dapat dijadikan referensi untuk membuat karya tulis.

Bagi Masyarakat:

1. SKOPIN dapat dimanfaatkan oleh masyarakat untuk menghindari kerusakan pada baterai perangkat elektronik yang terlalu lama diisi daya.
2. Memberikan kesadaran kepada masyarakat untuk menghemat listrik serta memanfaatkan pemakaian listrik secara tepat agar dapat mengurangi dan mencegah pemanasan global serta korsleting pada komponen yang berhubungan dengan listrik.

BAB II

KAJIAN PUSTAKA

A. Arduino

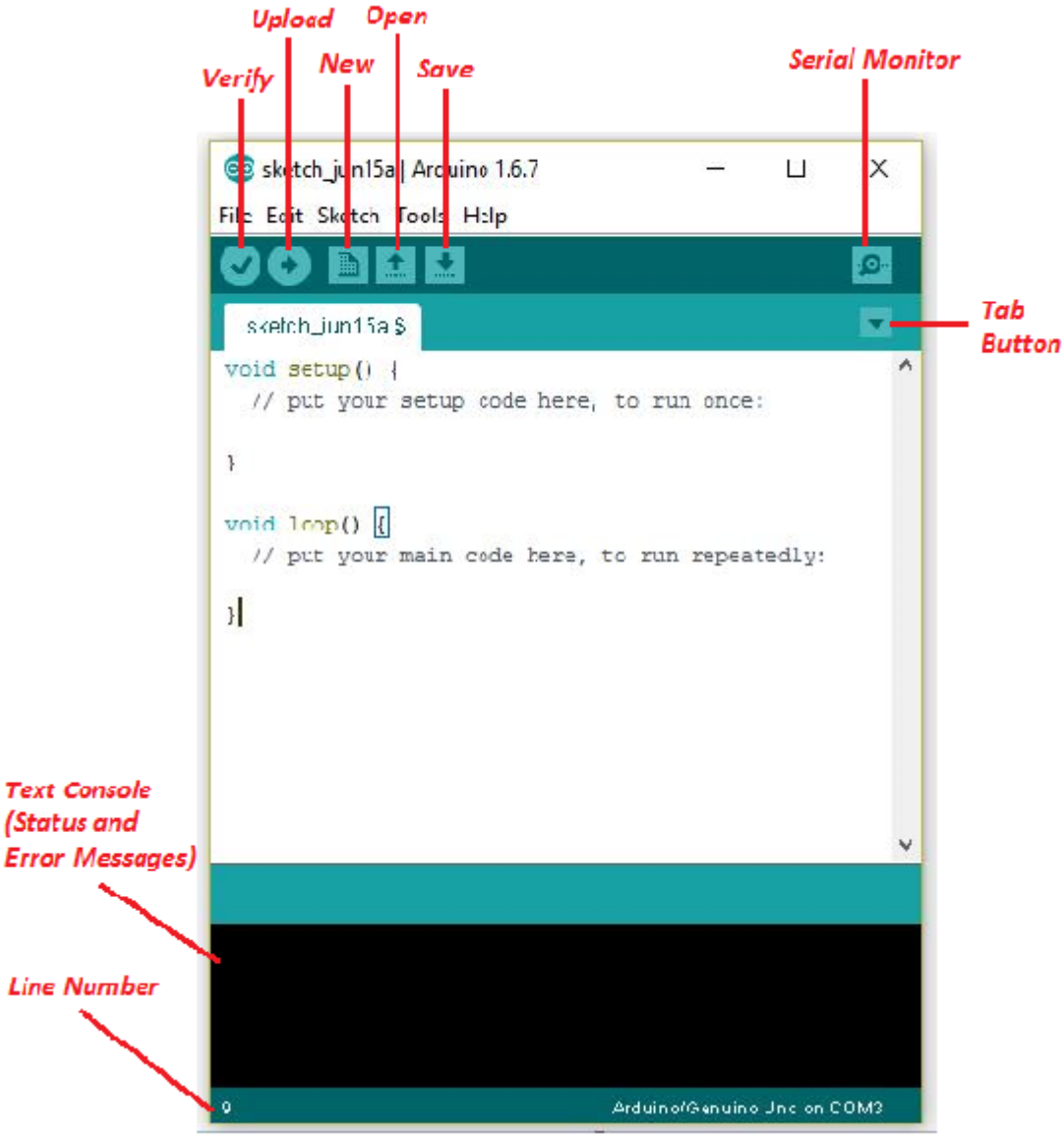
Mikrokontroler (*embedded controller*) adalah sistem yang mengandung masukan/keluaran, memori, dan prosesor, yang digunakan pada produk seperti mesin cuci, pemutar video, mobil, dan telepon [1]. Pada prinsipnya, mikrokontroler adalah sebuah komputer berukuran kecil yang dapat digunakan untuk mengambil keputusan, melakukan hal-hal yang bersifat berulang, dan dapat berinteraksi dengan peranti-peranti eksternal. Arduino adalah jenis suatu papan (*board*) yang berisi mikrokontroler. Dengan kata lain, Arduino dapat disebut sebagai sebuah papan mikrokontroler.

Arduino juga merupakan *platform hardware* terbuka (*open-source*) yang ditujukan kepada siapa saja yang ingin mengembangkan dan membuat berbagai peralatan elektronik interaktif berdasarkan *hardware* dan *software* yang fleksibel dan mudah digunakan. Menggunakan bahasa pemrograman arduino yang memiliki kemiripan syntax dengan bahasa pemrograman C. Karena sifatnya yang terbuka, maka siapa saja dapat mengunduh dan mengembangkan skema *hardware* arduino [14].

1. Arduino *Software*

Perangkat lunak program yang biasa disebut *sketch* dibuat pada komputer menggunakan Arduino *Integrated Development Environment* (IDE). Arduino dapat digunakan untuk menulis dan mengedit kode serta mengubah kode tersebut menjadi instruksi-instruksi yang dapat dimengerti oleh *hardware*. IDE juga memindahkan instruksi tersebut ke papan Arduino, prosesnya biasa disebut dengan *uploading*. Berikut adalah tampilan Arduino IDE.

Gambar 2.1: Tampilan ikon aplikasi *software* Arduino IDE



Gambar 2.2: Tampilan aplikasi *software* Arduino IDE

- *Verify* - Sebelum program "kode" yang telah dibuat dapat dikirim ke papan Arduino, perlu diubah menjadi petunjuk yang dipahami. Proses ini disebut dengan *compile*.
- *Upload* - Tombol *upload* akan mengkompilasi kemudian mentransmisikan instruksi tersebut melalui kabel USB ke papan Arduino.
- *New* - Membuka jendela baru untuk membuat *sketch* yang baru.
- *Open* - Memuat *sketch* dari sebuah file yang sudah tersimpan dalam komputer.
- *Save* - Ini akan menyimpan perubahan pada sketsa yang sedang dikerjakan.
- *Serial Monitor* - Sebuah fitur yang berfungsi sebagai tampilan Arduino pada komputer dengan menghubungkan Arduino dengan komputer menggunakan kabel USB dan port 0 dan port 1 pada Arduino tidak digunakan untuk keperluan lain.
- *Tab Button* - Ini memuat beberapa pilihan untuk membuat *sketch* baru, menganti nama *sketch*, menghapus *sketch*, membuka tab sebelumnya, membuka tab setelahnya.
- *Sketch Editor* - Tempat dimana dapat menulis dan mengedit *sketch*.
- *Text Console* - Ini menunjukkan apa yang sedang dilakukan IDE dan juga dimana pesan kesalahan muncul jika membuat kesalahan dalam mengetik program (Sering disebut syntax error).
- *Line Number* - Ini menunjukkan nomor baris yang berada pada kursor. Hal ini berguna karena *compiler* memberikan pesan kesalahan dengan nomor baris.

2. Arduino *Hardware*

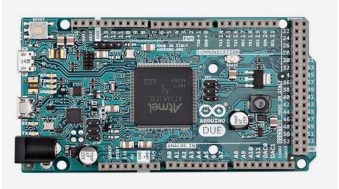
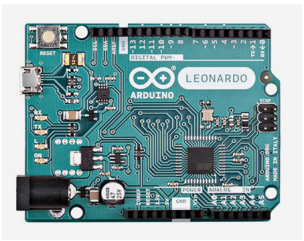
Arduino terdiri dari berbagai macam jenis, yaitu Arduino Uno, Arduino Duemilanove, Arduno Leonardo, Arduino Mega2560, Arduino Intel Galile, Arduino Pro Micro AT, Arduino Nano R3, Arduino mini Atmega, Arduino Mega ADK, dan Arduino Esplora. Penulis menggunakan Arduino Uno dalam penelitian ini dikarenakan Arduino Uno adalah mikrokontroler yang paling disarankan untuk digunakan oleh pemula, selain itu ukurannya yang pas dan harganya yang terjangkau membuat Arduino Uno juga menjadi jenis Arduino yang paling digemari masyarakat.

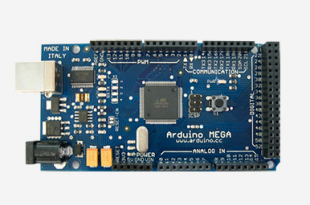
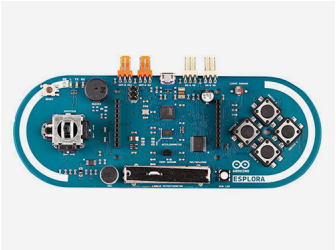
Arduino Uno Revision 3 merupakan versi terakhir dari Arduino Uno yang sekarang sedang marak digunakan karena memiliki banyak keunggulan dibandingkan dengan versi sebelumnya [8]. Arduino Uno R3 adalah *board* sistem minimum berbasis mikrokontroler ATmega328P jenis AVR. Arduino Uno R3 memiliki karakteristik sebagai berikut,

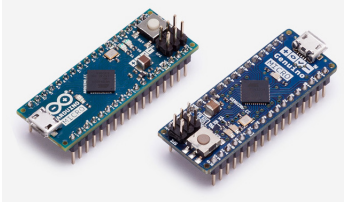
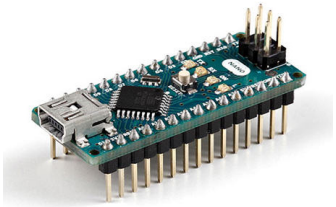
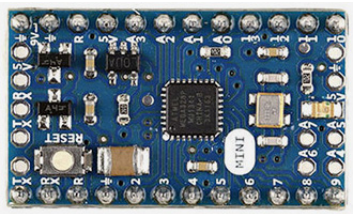
Tegangan Operasi	5 VDC
Rekomendasi Tegangan	7-12 VDC
Batas Tegangan	6-20 VDC
Jumlah I/O Digital	14 buah
Jumlah I/O Analog	6 buah
Arus DC setiap pin I/O	40mA
Arus DC untuk pin 3.3V	50mA
Flash memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Tabel 2.1: Spesifikasi Arduino UNO R3

Berikut ini adalah penjelasan singkat mengenai macam-macam Arduino disertai dengan gambar:

Board Arduino	Keterangan
	Arduino Uno memuat semua yang dibutuhkan untuk menunjang mikrokontroler, sehingga mudah dihubungkan ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya [9].
	**Arduino Due** adalah papan arduino pertama yang berdasarkan 32-bit ARM mikrokontroler. Dengan 54 digital input/output pin, 12 analog input, arduino due merupakan papan yang sempurna untuk projek arduino yang berskala lebih besar.
	**Arduino Leonardo** merupakan papan mikrokontroler berdasarkan ATmega32u4. Leonardo memiliki 20 digital pin input / output (7 diantaranya dapat digunakan sebagai output PWM dan 12 sebagai input analog), osilator kristal 16 MHz, koneksi micro USB, jack listrik, header ICSP, dan tombol reset.

	**Arduino Mega 2560** Memiliki 54 digital pin input / output (15 output, 16 analog input, 4 UART (hardware port serial)), osilator kristal 16 MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Kompatibel dengan sebagian besar *shield*, dirancang untuk Arduino Duemilanove atau Diecimila. Mega 2560 adalah *update* dari Arduino Mega.
	Arduino Pro adalah papan mikrokontroler berdasarkan ATmega328. Pro hadir dalam versi 3.3V / 8MHz dan 5V / 16MHz. Arduino Pro ditujukan untuk pemasangan semi permanen di objek karena tidak ada *header* yang sudah terpasang sebelumnya, yang memungkinkan penggunaan berbagai jenis konektor atau penyolderan langsung dengan kabel. Versi 3.3V Pro dapat didukung dengan baterai.
	**Arduino Esplora** adalah papan mikrokontroler berasal dari Arduino Leonardo. Esplora berbeda dari semua papan Arduino sebelumnya dalam hal ini menyediakan sejumlah built-in, dan dapat menggunakan sensor. Dirancang untuk orang yang ingin *build and run* dengan Arduino tanpa harus belajar tentang elektronik sebelumnya.

	**Arduino Mikro** adalah papan mikrokontroler berdasarkan ATmega32u4. Memiliki 20 digital pin *input* / *output*, osilator 16 MHz kristal, koneksi USB mikro, header ICSP, dan tombol reset. Berisi semua yang diperlukan untuk mendukung mikrokontroler; menghubungkannya ke komputer dengan kabel USB mikro untuk memulainya. Memiliki faktor bentuk yang memungkinkannya untuk dapat dengan mudah ditempatkan pada *breadboard*.
	**Arduino Nano** adalah sebuah papan kecil, lengkap, dan *beardboard-friendly* berdasarkan ATmega328. Memiliki fungsi yang hampir sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino nano tidak memiliki colokan listrik DC, dan bekerja dengan kabel USB Mini-B (bukan standar kabel yang biasa digunakan).
	**Arduino Mini** adalah papan mikrokontroler kecil yang memiliki 14 digital *input/output pins* (6 *output* digital, 8 *input* analog) dan 16 MHz *crystal oscillator*. Dapat di program dengan USB Serial adapter atau USB lainnya atau RS232 ke TTL serial adapter. Maksimal tegangan yang dapat diterima oleh Arduino mini adalah 9 volt.

3. *Library* Arduino

Sama seperti pada pemrograman menggunakan bahasa C/C++, pada bahasa Arduino kita dapat membuat *library* untuk memudahkan pemrograman. Arduino IDE menyediakan beberapa *library* yang dapat digunakan untuk memrogram Arduino dengan lebih mudah. *Library* pada dasarnya adalah kumpulan fungsi-fungsi yang dapat melakukan dan memudahkan suatu pekerjaan tertentu. Sebagai contoh, *built-in LiquidCrystal library* membuatnya mudah untuk berinteraksi dengan *display* LCD karakter.

Berikut ini penulis akan menjelaskan secara singkat beberapa *library* Arduino yang digunakan pada penerapan SKOPIN.

3.1 Countimer.cpp Berikut ini adalah *library* sederhana timer dan counter Arduino. Menawarkan tiga model pekerjaan:

a. *Count-up timer* dengan metode panggilan yang ditentukan saat penghitungan selesai.

b. *Count-down timer* dengan metode panggilan yang ditentukan saat perhitungan selesai.

c. Memanggil metode apapun pada interval waktu yang ditentukan.

Berikut ini adalah metode yang digunakan untuk menjalankan aksi pada SKOPIN:

a. void start()

   Fungsi untuk memulai *timer*.

b. byte getCurrentHours()

   Fungsi untuk mendapatkan waktu dalam jam terbaru pada *timer*.

c. byte getCurrentMinutes()

Fungsi untuk mendapatkan waktu dalam menit terbaru pada *timer*.

d. byte getCurrentSeconds()

Fungsi untuk mendapatkan waktu dalam detik terbaru pada *timer*.

e. void setCounter()

Fungsi untuk mengatur waktu (jam, menit, detik) untuk *timer* yang sudah ada.

f. void setInterval()

Fungsi untuk menjalankan fungsi lain yang sudah ditentukan pada interval tertentu (dalam milidetik).

g. String getCurrentTime()

Fungsi untuk mendapatkan waktu terbaru *timer* dalam format HH:MM:SS.

h. bool isCounterCompleted()

Fungsi akan *return True* apabila perhitungan selesai dan *return False* apabila belum selesai.

i. bool isCounterRunning()

Fungsi akan *return True* apabila perhitungan masih berjalan dan *return False* apabila belum selesai.

j. bool isStopped()

Fungsi akan *return True* apabila *timer* berhenti dan *return False* apabila tidak berhenti.

3.2 LiquidCrystal.cpp

a. LiquidCrystal()

Membuat variabel tipe *LiquidCrystal*. Layar dapat dikontrol menggunakan 4 atau

8 jalur data. Jika yang pertama, hilangkan nomor pin untuk d0 ke d3 dan biarkan garis tersebut tidak terhubung. Pin RW dapat dihubungkan ke *ground* dan bukan terhubung ke pin pada Arduino; Jika demikian, hilangkan dari parameter fungsi ini.

b. begin()

Menginisialisasi antarmuka ke layar LCD, dan menentukan dimensi (lebar dan tinggi) layar. begin() perlu dipanggil sebelum perintah *library* LCD lainnya.

c. clear()

Membersihkan layar LCD dan posisi kursor terletak di sudut kiri atas.

d. home()

Posisi kursor terletak di kiri atas LCD. Dapat menggunakan lokasi tersebut untuk menampilkan teks berikutnya ke layar.

e. setCursor()

Memposisikan kursor LCD. Dapat mengatur lokasi penulisan teks berikutnya ke LCD yang akan ditampilkan, dengan *syntax* "*lcd.setCursor(col, row)* "

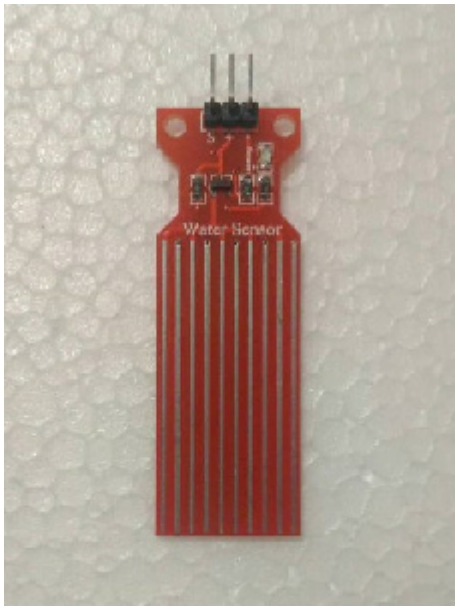
f. print()

Fungsi *print* adalah untuk menampilkan tulisan ke LCD.

B. Sensor Air

Sensor air (*Water level sensor*) dirancang untuk mendeteksi air, yang dapat secara luas digunakan dalam curah hujan penginderaan, tingkat air, dan bahkan kebocoran cairan [13]. Berbeda dengan relay yang menggunakan sinyal digital, modul sensor air menggunakan sinyal analog untuk mengirimkan data kepada arduino. Data analog yang dikirimkan adalah dari *range* 0 - 1024.

Sensor air digunakan sebagai fitur tambahan untuk melengkapi SKOPIN. Sensor air berfungsi sebagai pendeteksi air disekitar SKOPIN, misalnya terkena air secara tidak sengaja ataupun terjadi bencana alam (banjir). Apabila sensor sudah terkena air dengan batas yang telah ditentukan, maka sensor akan mengirimkan sinyal analog ke Arduino dan Arduino akan menampilkan perintah kepada pengguna melalui LCD untuk mematikan / me-*reset* alat, selain itu apabila *timer* sedang berjalan maka secara otomatis arus akan diputus oleh Arduino.



Gambar 2.3: Sensor air

C. Relay

Relay adalah sebuah saklar yang dikendalikan oleh arus. Saklar adalah sebuah perangkat yang digunakan untuk memutuskan atau menyambungkan aliran listrik. Relay memiliki sebuah inti. Terdapat sebuah armatur besi yang akan tertarik menuju inti apabila arus yang mengalir melewati kumparan. Artmatur ini terpasang pada sebuah tuas pegas. Ketika artmatur tertarik menuju inti, kotak jalur yang bersamaan akan berubah posisinya dari kontak normal tertutup ke kontak normal terbuka [3].

Konstruksi dalam suatu relay terdiri dari lilitan kawat (coil) yang dililitkan pada inti besi lunak. Jika lilitan kawat mendapatkan aliran arus, inti besi lunak kontak menghasilkan medan magnet dan menarik switch kontak. Dan pemakaian jenis relay tergantung pada keadaan yang diinginkan dalam suatu rangkaian. Salah satu bentuk relay dapat dilihat pada gambar 3.

Gambar 2.4: Relay yang akan digunakan yaitu 2 Relay Module.

Menurut kerjanya relay dapat dibedakan menjadi:

1. *Normaly Open (ON)*, saklar akan terbuka bila dialiri arus
2. *Normaly Close (OFF)*, saklar akan tertutup bila dialiri arus
3. *Change Over (CO)*, relay ini mempunyai saklar tunggal yang normalnya tertutup lama, bila kumparan 1 dialiri arus maka saklar akan terhubung ke terminal A, sebaliknya bila kumparan 2 dialiri arus maka saklar akan terhubung ke terminal B

Relay memiliki manfaat, yaitu;

1. Dapat mengontrol sendiri arus serta tegangan listrik yang diinginkan
2. Dapat memaksimalkan besarnya tegangan listrik hingga mencapai batas maksimalnya

3. Dapat menggunakan baik saklar maupun koil lebih dari satu, disesuaikan dengan kebutuhan

D. LCD (Liquid Crystal Display)

LCD (Liquid Crystal Display) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (Liquid Cristal Display) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari back-lit. LCD terdiri dari sel-sel. Sel-sel pada LCD terbuat dari pembagian dua kaca yang mengandung kristal cair kecil yang berada pada layer didalamnya [4]. LCD berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik [2].

Gambar 2.5: LCD yang akan digunakan yaitu Modul LCD Keypad.

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan

elektroda transparan indium oksida dalam bentuk tampilan seven-segment dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan sandwich memiliki polarizer cahaya vertikal depan dan polarizer cahaya horisontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.

1. Pengendali / Kontroler LCD (Liquid Cristal Display)

Dalam modul LCD terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan register. Memori yang digunakan mikrokontroler internal LCD adalah:

1. DDRAM (Display Data Random Access Memory) merupakan memori tempat karakter yang akan ditampilkan berada.

2. CGRAM (Character Generator Random Access Memory) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.

3. CGROM (Character Generator Read Only Memory) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD tersebut sehingga pengguna tinggal mangambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.

Register control yang terdapat dalam suatu LCD diantaranya adalah

a. Register perintah yaitu register yang berisi perintah-perintah dari mikrokontroler ke panel pada saat proses penulisan data atau tempat status dari panel LCD dapat dibaca pada saat pembacaan data.

b. Register data yaitu register untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

Pin, kaki atau jalur input dan kontrol dalam suatu LCD diantaranya adalah

1. Pin data adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.

2. Pin RS *(Register Select)* berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukkan yang masuk adalah perintah, sedangkan logika *high* menunjukkan data.

3. Pin R/W *(Read Write)* berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* baca data.

4. Pin E *(Enable)* digunakan untuk memegang data baik masuk atau keluar.

5. Pin VLCD berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke *ground*, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

E. FSM (*Finite State Machine*)

Finite State Machines (FSM) adalah sebuah sistem reaktif yang menanggapi stimulus tertentu, dapat berupa sinyal atau sepotong input yang menggambarkan cara

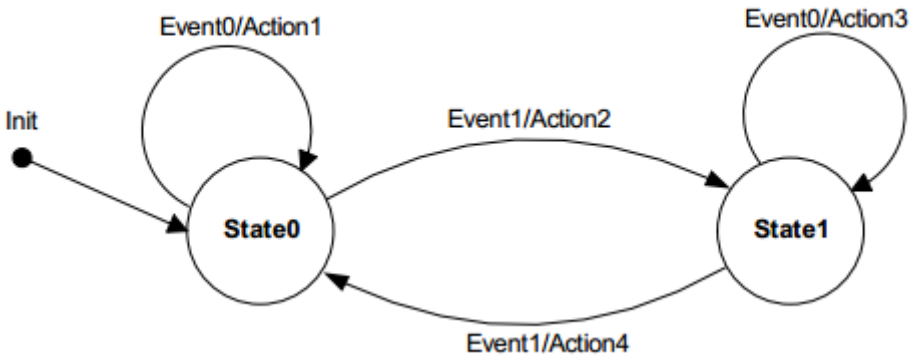
kerja sistem dengan menggunakan tiga kondisi berikut: *State* (Keadaan), *Event* (kejadian) dan *Action* (aksi). Pada saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks [10].

Metode FSM sangat cocok digunakan sebagai basis perancangan perangkat lunak pengendalian yang bersifat reaktif dan *real time*. Salah satu keutungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relatif besar dengan hanya menggunakan sejumlah kecil *item state*. Selain untuk bidang kontrol, Penggunaan metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak *game*, aplikasi WEB dan sebagainya.

1. Diagram Keadaan

Diagram keadaan pada dasarnya merupakan salah satu bentuk representasi dari FSM. Diagram ini secara visual menggambarkan tingkah laku yang dimiliki oleh sistem kontrol yang kompleks kedalam bentuk yang lebih sederhana dan relatif mudah dipahami. Dalam diagram ini, *state-state* yang terdapat pada sebuah sistem digambarkan sebagai lingkaran yang diberi label unik, sedangkan transisi *state* yang diakibatkan oleh *event* tertentu direpresentasikan sebagai anak panah yang berasal dari *state* yang ditinggalkan menuju *state* yang aktif. Setiap transisi yang terjadi umumnya juga diikuti oleh aksi yang dilakukan oleh sistem yang dirancang. Secara praktis setiap diagram state yang dirancang akan selalu memiliki sebuah transisi awal

(inisial) yang menuju salah satu *state* sejak sistem kontrol tersebut mulai dihidupkan. Gambar berikut memperlihatkan contoh penggambaran diagram *state*:



Gambar 2.6: Contoh diagram *state*.

Diagram tersebut memperlihatkan FSM dengan dua buah *state* dan dua buah *input* serta empat buah aksi *output* yang berbeda: seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju state0, pada keadaan ini sistem akan menghasilkan Action1 jika terjadi masukan Event0, sedangkan jika terjadi Event1 maka Action2 akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan State1 dan seterusnya.

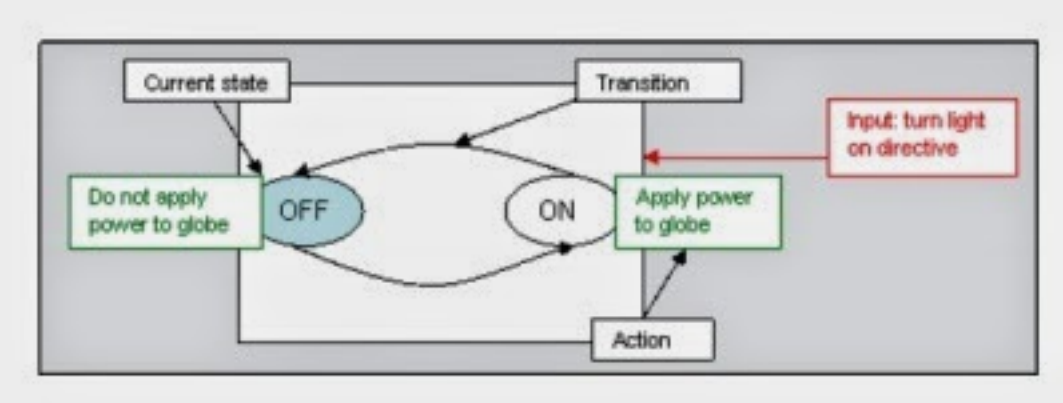
Algoritma E..1 Potongan Program Pada Gambar 2.6

```
...
#difine TRUE 1
enum {state0,state1}
...
unsigned char state;
void Action1(void);
void Action2(void);
void Action3(void);
void Action4(void);
```

```
...
main()
{
//init
...
While(TRUE)
 {
  switch(state)
  {
    case state0:if(input ==Event0)
            Action1();
          If(input==Event1)
            {Action2();
            state=state1;
            }
            break;
    case state1:if(input ==Event0)
            Action3();
          If(input==Event1)
            {Action4()
            state=State0;
              }
                break;
  }
 }
...
}
```

Ada dua metode utama untuk memperlakukan FSM untuk menghasilkan output. Yaitu *Moore Machine* dan *Mealy Machine* yang dinamakan berdasarkan penemunya.

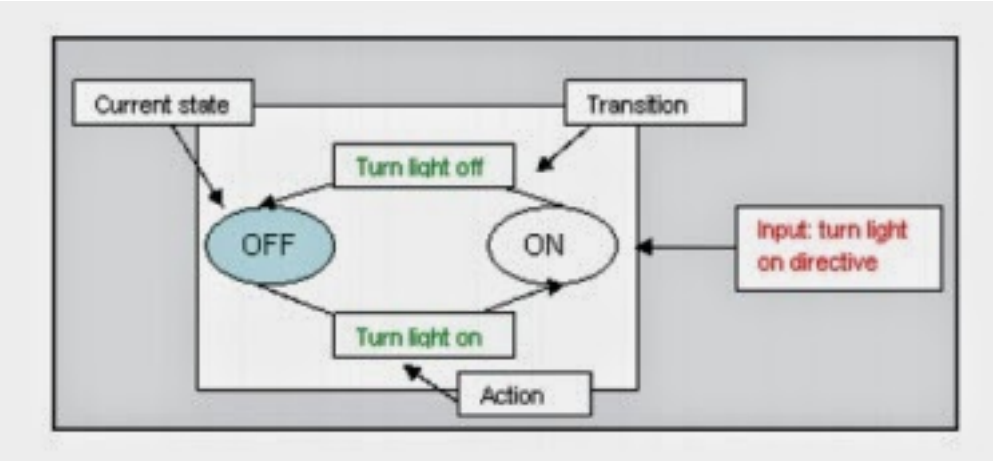
a. *Moore Machine*



Gambar 2.7: *Moore State Machine*.

Moore Machine adalah tipe dari FSM dimana *output* dihasilkan dari *state*. Pada gambar di atas mencontohkan dimana *state* mendefinisikan apa yang harus dilakukan. Keluaran pada *Moore Machine* diasosiasikan sebagai *state*. Dan pada penelitian ini, penulis menggunakan Moore Machine

b. *Mealy Machine*



Gambar 2.8: *Mealy State Machine*.

Mealy Machine berbeda dengan *Moore Machine* dimana keluarannya merupakan hasil dari transisi antar *state*. Keluaran pada *Mealy Machine* diasosiasikan sebagai tran-

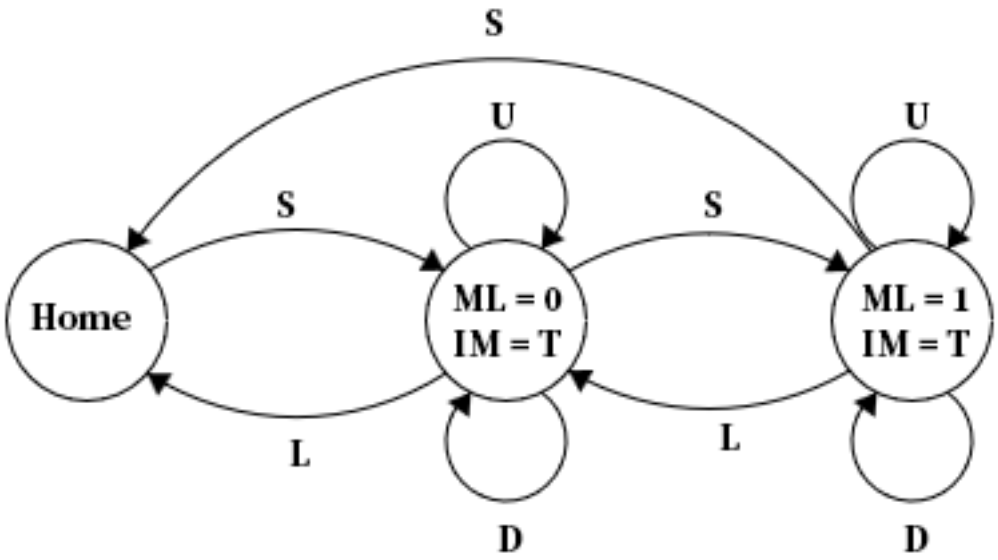
sisi.

Berikut adalah tabel perbedaan pada *Mealy* dan *Moore Machine*.

Mealy Machine	*Moore Machine*
Output tergantung pada keadaan sekarang dan masukan sekarang	Output hanya bergantung pada keadaan sekarang.
Umumnya, ia memiliki *state* lebih sedikit daripada *Moore Machine*	Umumnya, ia memiliki lebih banyak *state* daripada *Mealy Machine*
Mealy machine bereaksi lebih cepat terhadap input	Dalam *Moore machine*, lebih banyak membutuhkan logika untuk memecahkan kode keluaran karena memiliki *delay* sirkuit yang lebih banyak

Tabel 2.2: Perbedaan *Mealy dan Moore Machine*

Berikut adalah rancangan diagram untuk mengatur / konfigurasi cara kerja yang akan diimplementasikan pada SKOPIN



Gambar 2.9: Diagram rancang bangun FSM SKOPIN.

Diagram FSM di atas dibagi kedalam 3 keadaan, yaitu keadaan saat tampilan di beranda (*home*), pada saat memilih menu, dan pada saat mengatur waktu. Setiap keadaan ditentukan oleh beberapa *flags* pada variabel yang ditentukan, untuk lebih jelasnya berikut ini adalah tabel keadaan dari diagram FSM di atas:

Variabel / Keadaan	Home	Pilih Menu	Atur Waktu
Menu Level	0	0	1
Is Menu	0	1	1

Tabel 2.3: Tabel Keadaan

Berikut ini penjelasan masing-masing keadaan:

1. *Home*

   Pada keadaan ini, tampilan alat berada pada tampilan paling awal yang menampilkan informasi terminal (stop kontak) mana yang sedang aktif.

2. Pilih Menu

Pada keadaan ini, sudah berada didalam menu untuk memilih terminal yang ingin diatur waktunya

3. Atur Waktu

   Pengguna akan diminta untuk memasukkan waktu yang diinginkan dengan menekan tombol *Up* atau *Down*.

Setiap *event* yang dilakukan akan mempengaruhi kondisi *flags* pada variabel yang menentukan keadaanya. Terdapat 4 variabel yang dipengaruhi oleh *event Up*, *Down*, *Left*, dan *Select* yaitu *Menu Level, Is Menu, Menu Page*, dan *Timer Count*. Berikut ini penjelasan dari masing-masing variabel:

1. *Is Menu*

   Variabel ini menentukan posisi sedang berada di *home* atau tidak.

2. *Menu Level*

   Variabel ini menentukan posisi menu berada pada *parent* atau *child*.

3. *Menu Page*

   Varibel ini untuk menentukan posisi menu untuk menyeleksi terminal yang ingin dipilih.

4. *Timer Count*

   Variabel ini digunakan sebagai lokasi sementara untuk mengatur *timer* dari terminal yang ingin digunakan.

Berikut ini adalah detail perubahan yang dilakukan pada setiap *event*:

State\ Event	Current Flags	Next Event			
		Up	Down	Left	Select
Home	ML = 0 IM = 0	-	-	-	ML = 0 IM = 1
Pilih Menu	ML = 0 IM = 1	ML = 0 IM = 1 MP--	ML = 0 IM =1 MP++	ML = 0 IM = 0	ML = 1 IM = 1
Atur Waktu	ML =1 IM = 1	ML = 1 IM = 1 TC++	ML = 1 IM =1 TC--	ML = 0 IM = 1	ML = 0 IM = 0

Tabel 2.4: Penejelasan diagram FSM.

BAB III

IMPLEMENTASI PROGRAM

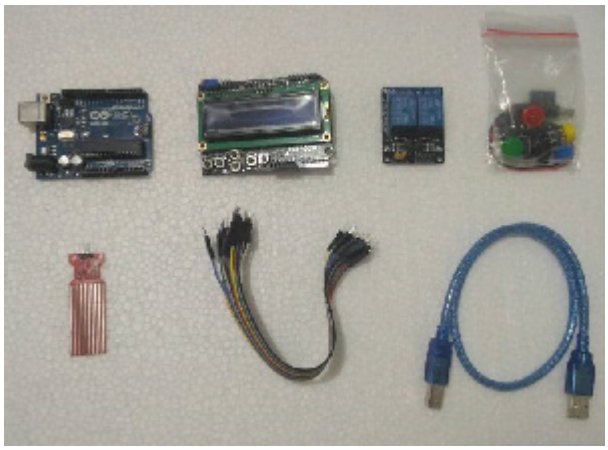
A. Tempat dan Waktu Penelitian

Penelitian dan pengimplementasian alat dilaksanakan di rumah penulis beralamat di Jalan Masjid No.27 Susukan, Ciracas, Jakarta Timur. Waktu penelitian berlangsung selama kurang lebih 5 bulan, mulai dari bulan Februari sampai dengan bulan Juli 2017.

B. Metode Penelitian

1. Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian ini terbagi atas perangkat keras dan perangkat lunak yang akan dijelaskan sebagai berikut.



Gambar 3.1: Alat dan bahan yang digunakan untuk pembuatan skopin

1.1 Perangkat Keras Penulis menggunakan beberapa perangkat keras *(hardware)* yang akan digunakan untuk membuat SKOPIN. Berikut adalah komponen yang akan

digunakan.

a. Arduino Uno R3 (1 unit)

   Arduino digunakan sebagai mikrokontroler yang berfungsi sebagai alat pengendali sekaligus alat yang menghubungkan perangkat-perangkat agar dapat menjalankan fungsinya.

b. Moodul LCD *Keypad* (1 unit)

   Keypad berfungsi untuk memberikan input perintah yang akan ditampilkan pada LCD berapa lama waktu yang akan diatur untuk mengaliri listrik

c. 2 Relay Module (1 unit)

   Relay Modul ini berfungsi sebagai saklar atau switch ganda yaitu untuk memutuskan atau meneruskan arus. Relay akan dihubungkan dengan stop kontak dan diberikan perintah oleh mikrokontroler Arduino.

d. Terminal Kuningan 2 Slot (1 unit)

   Terminal kuningan merupakan stop kontak yang akan diintegerasikan dengan pengatur waktu.

e. Sensor Air (1 unit)

   Sebagai fitur tambahan pada SKOPIN, yang berfungsi untuk melindungi SKOPIN dari kerusakan akibat terkena air.

f. Kabel Listrik (2 unit)

   Berfungsi untuk menghubungan SKOPIN dengan arus listrik.

g. Kabel Jumper (1 unit)

   Berfungsi untuk menghubungkan antara Arduino, relay, dan LCD.

h. Kabel USB (1 unit)

   Berfungsi untuk menghubungkan SKOPIN dengan laptop untuk kemudian meng-*upload* program dari Arduino IDE ke SKOPIN.

i. Solder dan Timah

Digunakan untuk membuat dan menghubungkan semua perangkat menjadi sebuah alat yang utuh.

j. Baterai dan Powerbank (2 dan 1 unit)

Alat penunjang agar SKOPIN dapat menyala

k. Tombol *On/Off* (1 unit)

Untuk menghidupkan dan Mematikan SKOPIN

l. *Breadboard* (1 unit)

Untuk menghubungkan komponen dan memberikan cabang untuk penggunaan kabel jumper.

m. Baut, Spacer, Tang, Obeng

Alat bantu penunjang dalam merangkai komponen.

1.2 Perangkat Lunak Penulis juga menggunakan perangkat lunak *(software)* sebagai alat penunjang untuk membuat SKOPIN. Berikut adalah beberapa perangkat lunak yang digunakan.

a. Arduino IDE Windows

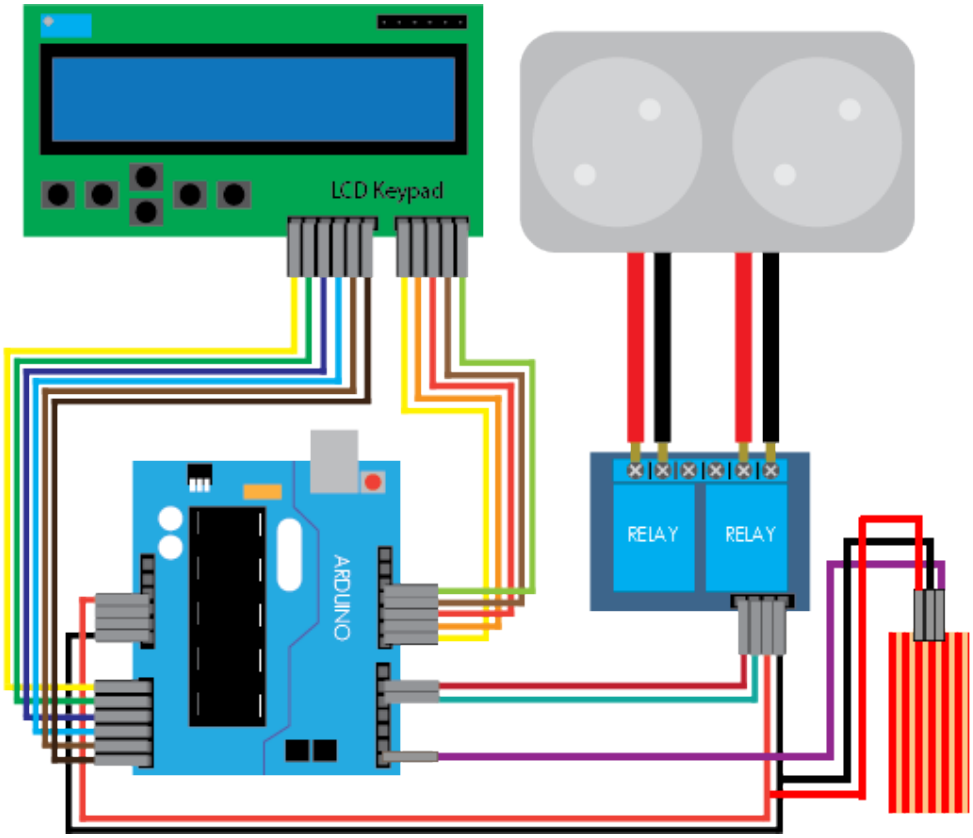
Perangkat lunak ini merupakan editor dan untuk mengirimkan instruksi berupa program menggunakan bahasa C++ ke dalam arduino

b. Adobe Illustrator dan Adobe Photoshop Penulis gunakan untuk membuat sketsa rangkaian alat

2. Prosedur Penelitian

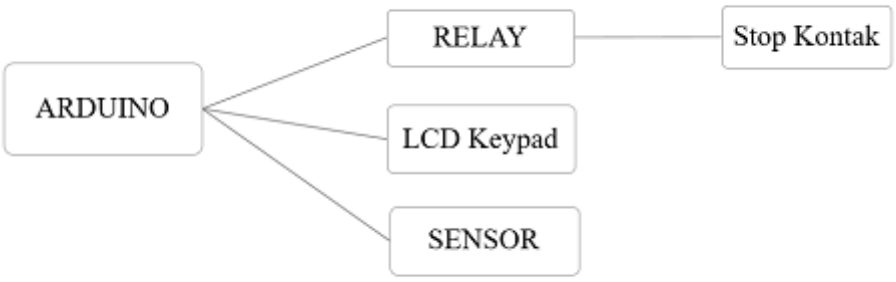
SKOPIN terdiri dari stop kontak sebagai penghubung steker dengan arus listrik. Agar SKOPIN dapat mengatur waktu untuk mengaliri arus listrik, maka penulis membuat program menggunakan metode FSM untuk memberikan prinsip *timer* pada

Arduino serta dihubungkan dengan keypad LCD untuk memberikan input masukan. Relay digunakan Untuk memutuskan dan meneruskan arus dengan aman. Berikut adalah gambar desain perancangan alat pada SKOPIN.



Gambar 3.2: Desain Rancangan Alat.

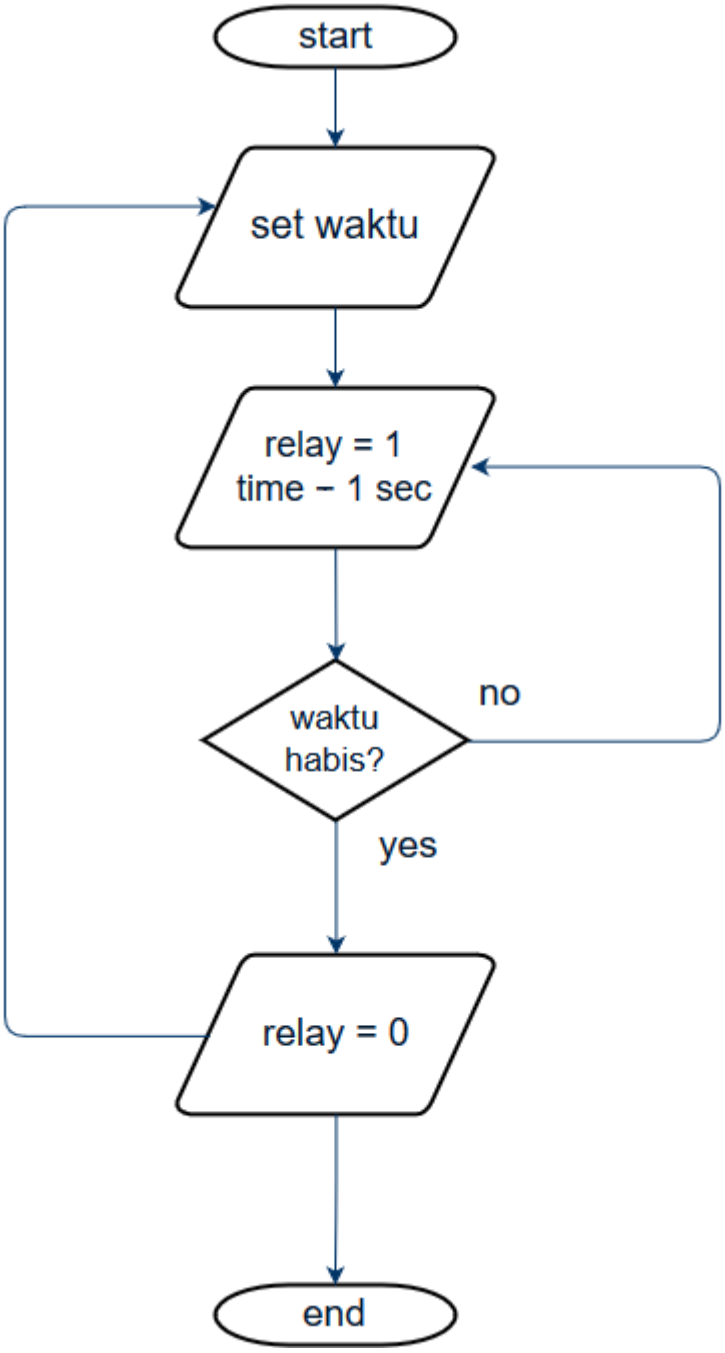
Berikut adalah diagram rancangan alat SKOPIN.



Gambar 3.3: Diagram Rancangan Alat.

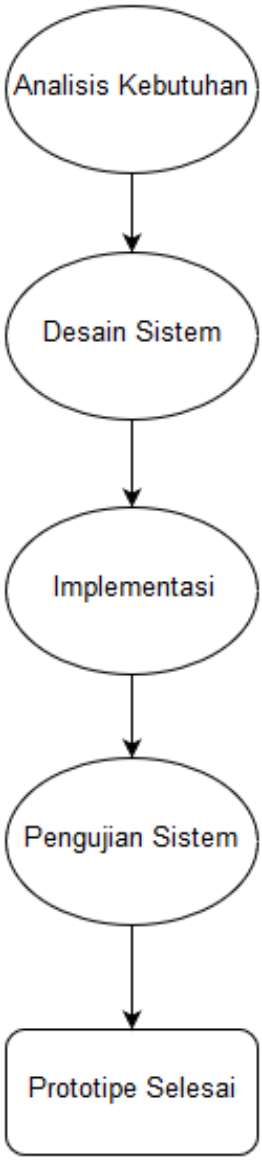
2.1 Cara Kerja Alat

1. Hubungkan steker dengan terminal pada SKOPIN,
2. Tekan tombol *ON* untuk menyalakan LCD,
3. Tekan tombol *Select*, lalu dengan menekan tombol ke atas atau ke bawah, pilih terminal mana yang akan digunakan (Terminal A atau B), lalu tekan *Select*,
4. Pilih menu berapa lama waktu yang akan diatur untuk mengaliri listrik menggunakan tombol ke atas atau ke bawah, lalu tekan *Select*,
5. Jika sudah menekan tombol *Select*, akan muncul tulisan “Set, Ok! ”kemudian SKOPIN akan menghitung mundur dari waktu yang sudah diatur dan aliran listrik akan terputus sampai waktu yang sudah ditentukan.
6. Apabila timer sedang berjalan, lalu sensor terkena air dengan jumlah yang ditentukan maka akan muncul tampilan di layar “Terdeteksi Air! silahkan matikan perangkat ”lalu pada saat yang bersamaan Arduino akan memutuskan arus listrik yang sedang mengalir.



Gambar 3.4: Diagram alur SKOPIN.

C. Teknik Pengumpulan dan Analisis Data



Gambar 3.5: Diagram Tahapan Pelaksanaan.

a. Analisis Kebutuhan

Tahapan pertama yang dilakukan adalah mengidentifikasi dan mendapatkan data mengenai kebutuhan apa saja yang diperlukan dalam perancangan dan pengimplementasian sistem serta pemikiran untuk perancangan selanjutnya.

b. Desain Sistem

Tahapan kedua yang dilakukan adalah membuat gambaran mengenai sistem perancangan model yang akan dibuat. Kemudian membuat alur *flowchart* sebagai gambaran bagaimana proses yang dilakukan untuk menggunakan alat. Selain itu, membuat desain tampilan yang bertujuan agar pengguna dapat memahami cara memilih waktu yang tepat untuk mengatur lamanya aliran listrik pada LCD. Terakhir, membuat sebuah prototipe selain mudah untuk digunakan juga harus didesain secara baik dan rapi agar dapat menarik untuk digunakan.

c. Implementasi

Tahap implementasi merupakan tahap dimana penulis menerjemahkan modul-modul desain sistem ke dalam bentuk aplikasi dengan menggunakan bahasa pemrograman C++ ke dalam Android IDE Windows. Dengan kata lain, penulis akan memprogram mikrokontroler agar sistem pada SKOPIN dapat berfungsi dengan baik.

d. Pengujian Sistem

Pengujian *debugging* merupakan proses eksekusi pada program untuk menemukan kesalahan. Sebelum program diterima oleh pengguna, maka program harus dapat terlepas dari kesalahan implementasi. Oleh sebab itu aplikasi harus diuji untuk menemukan berbagai kesalahan yang mungkin terjadi. Setelah melakukan pengujian dan telah mendeteksi adanya kesalahan, setelah itu harus diperbaiki sampai prototipe benar-benar selesai.

BAB IV

PERCOBAAN DAN HASIL PERCOBAAN

A. Tahapan Uji Coba

Setelah proses perakitan alat dan pemrograman, tahapan berikutnya adalah uji coba alat. Pengujian dimaksudkan untuk mengetahui apakah alat tersebut dapat bekerja dengan baik dan sudah sesuai dengan rancangan yang dibuat. Berikut adalah hal-hal yang dilakukan sebelum melakukan uji coba:

1. Menyiapkan data-data pengujian yang dibutuhkan.
2. Menyiapkan *Smarthphone* untuk mengukur keakuratan waktu SKOPIN.
3. Wadah berisi air dan sedotan air mineral untuk mengetes sensor air pada SKOPIN.
4. Menyiapkan sumber arus AC untuk SKOPIN.

Berikut adalah tampilan alat (SKOPIN) yang telah dibuat oleh penulis dan akan dilakukan uji coba hasil.

Gambar 4.1: Tampilan SKOPIN dari atas

SKOPIN terdiri dari 2 terminal, yaitu terminal A dan terminal B. Terminal A diatur untuk melakukan percobaan singkat yaitu per 15 detik sampai 4 menit. Pada penerapannya, terminal A akan memiliki pengaturan waktu yang sama dengan terminal B yaitu mulai dari 15 menit dan berlaku kelipatannya hingga maksimal waktu 4 jam. Proses uji coba dilakukan terhadap 3 variabel, yaitu keakuratan waktu, fungsi *timer* pemutus arus dan sensor air. Berikut ini adalah data yang digunakan untuk menghitung keakuratan waktu SKOPIN dan fungsi timer pemutus arus:

1. Terminal A:

   Terminal A diatur untuk percobaan dalam waktu singkat. Minimal 15 detik dan maksimal adalah 240 detik atau 4 menit dengan pengaturan dapat menyetel waktu setiap kelipatan 15 dalam satuan detik.

Waktu dalam Detik			
15	30	45	60
75	90	105	120
135	150	165	180
195	210	225	240

Tabel 4.1: Data pengetesan keakuratan *timer* SKOPIN terminal A

2. Terminal B :

   Terminal B diatur untuk percobaan dalam waktu yang akan diterapkan. Minimal 15 menit dan maksimal adalah 240 menit atau 4 jam dengan pengaturan dapat menyetel waktu setiap kelipatan 15 dalam satuan menit. Data yang digunakan untuk percobaan pada terminal B menggunakan waktu yang berjarak 2 kali lipat dari data sebelumnya, dimulai dari 0,25 jam.

Waktu dalam Jam				
0,25	0,5	1	2	4

Tabel 4.2: Data pengetesan keakuratan *timer* SKOPIN terminal B

Tinggi batas sensor air yang dapat mendeteksi ketinggian air adalah 1,5 inch atau kurang lebih 3,8 cm. Berikut ini adalah perlakuan untuk pengujian sensor air:

1. Tetes Sedikit
2. Tetes Banyak
3. Celup 1 mm
4. Celup 0.5 cm
5. Celup 1 cm

6. Celup 2 cm
7. Celup 3 cm
8. Celup 3,8 cm

B. Hasil Percobaan

SKOPIN dilakukan percobaan dengan cara menjalankan alat tersebut sesuai dengan cara kerja alat, kemudian setelah mengatur waktu yang sesuai dengan data yang akan digunakan, secara bersamaan *stopwatch* pada *smartphone* akan digunakan untuk menghitung selisih waktu dengan SKOPIN.

Pada uji coba sensor air, percoban dilakukan dengan melakukan pentetesan air pada sensor dan pencelupan sensor ke dalam air. Hal ini dilakukan untuk melihat seberapa besar sensor mengetahui batasan air yang diinginkan untuk mematikan arus listrik. Penulis menginginkan agar arus mati ketika sensor mendeteksi terdapat tetesan air yang banyak atau pada ketinggian air 1 mm. Setelah itu dilakukan pengetesan pada setiap ketinggian air untuk mengetahui bahwa sensor mendeteksi batas air degan baik.

1. Hasil Percobaan Keakuratan Waktu & fungsi Timer

Berikut ini adalah hasil percobaan SKOPIN pada Terminal A:

No.	Setting Waktu	Timer Smartphone	Selisih Waktu	Selisih / Detik	Keterangan
1	15	00.00.15,13	+0,13	0.0087	berfungsi
2	30	00.00.30,22	+0,22	0,0073	berfungsi
3	45	00.00.45,26	+0,26	0,0057	berfungsi
4	60	00.01.00,77	+0,77	0,0128	berfungsi
5	75	00.01.16,06	+1,06	0,0141	berfungsi
6	90	00.01.31,31	+1,31	0,0145	berfungsi
7	105	00.01.46,33	+1,33	0,0127	berfungsi
8	120	00.02.01,75	+1,75	0,0146	berfungsi
9	135	00.02.16,87	+1,87	0,0139	berfungsi
10	150	00.02.31,99	+1,99	0,0133	berfungsi
11	165	00.02.47,12	+2,12	0,0128	berfungsi
12	180	00.03.02,57	+2,57	0,0143	berfungsi
13	195	00.03.17,60	+2,60	0,0133	berfungsi
14	210	00.03.32,80	+2,80	0,0133	berfungsi
15	225	00.03.48,20	+3,20	0,0142	berfungsi
16	240	00.04.03,86	+3,86	0,0161	berfungsi

Tabel 4.3: Hasil percobaan pada SKOPIN teriminal A

Berikut ini adalah hasil percobaan SKOPIN pada Terminal B

No.	Setting Waktu	Timer Smartphone	Selisih Waktu	Selisih / Menit	Keterangan
1	0,25	00.15.10,98	+10,98	0,7320	berfungsi
2	0,5	00.30.21,58	+21,58	0,7193	berfungsi
3	1	01.00.44,36	+44,36	0,7393	berfungsi
4	2	02.01.28,35	+1.28,35	0,7362	berfungsi
5	4	04.02.43,77	+2.43,77	0,6824	berfungsi

Tabel 4.4: Hasil percobaan pada SKOPIN terminal B

Berdasarkan hasil percobaan pada tabel di atas, penulis menyimpulkan bahwa waktu yang diatur untuk mengaliri listrik pada SKOPIN apabila dibandingkan dengan *stopwatch* pada *smartphone* memiliki selisih waktu. Semakin lama waktu yang diatur pada SKOPIN, semakin besar pula selisih waktu dengan *stopwatch* pada *smartphone*. Untuk memastikan keakuratan data pengujian, penulis mengambil sampel pada salah satu *timer* SKOPIN yaitu 15 detik untuk dites selisihnya sebanyak 30 kali. Hasil dari penjabaran pengujian ini penulis sertakan pada lampiran A. Berikut ada perhitungan rata-rata selisih waktu SKOPIN dengan *stopwatch* pada *smartphone*:

1. Terminal A:

   $\bar{x}$ = $\frac{(0.0087+0,0073+0,0057+0,0128+0,0141+0,0145+0,0127+0,0146+0,0139+0,0133+0,0128+}{16}$

   $\frac{0,0143+0,0133+0,0133+0,0142+0,0161)}{16} = 0,0116$ detik

2. Terminal B:

   $\bar{x}$ = $\frac{(0,7320+0,7193+0,7393+0,7362+0,6824)}{5} = 0,7218$ menit

Rata-rata selisih waktu antara SKOPIN dengan *stopwatch* pada *smartphone* untuk terminal A setiap detik adalah 0,0116 detik sedangkan untuk terminal B setiap menit adalah 0,7218 menit.

Berdasarkan hasil uji coba di atas, SKOPIN dapat berfungsi dengan baik untuk memutuskan arus listrik sesuai dengan waktu yang telah ditetapkan walaupun *charger* tidak dicabut langsung setelah baterai penuh oleh pengguna. Hal ini membuktikan bahwa SKOPIN dapat mengurangi penggunaan listrik berlebihan yang dilakukan oleh beberapa pengguna *smartphone* di Indonesia. Jika pengguna membiarkan perangkatnya diisi daya melebihi kapasitas, maka akan tetap menyerap listrik 2,24 watt setiap jam dan apabila meninggalkan *charger* terus menerus tanpa perangkat, akan tetap menyerap daya sebesar 0,26 watt setiap jam.

Berikut adalah biaya listrik jika perangkat dikendalikan dengan timer pada SKOPIN.

Daya saat baterai lemah = 3,68 Watt

Daya Arduino = 1,5 Watt

Harga listrik = Rp. 1.467,28 / KWh

Estimasi Waktu mengisi Daya hingga perangkat penuh = 2 jam

$\frac{(3,68x2)}{1000} x 1.467,28 =$ Rp. 10,7992

$\frac{(1,5x2)}{1000} x 1.467,28 =$ Rp. 4,4018

Rp. 10,7992 + Rp. 4,4018 = Rp. 15,2010

Jika ditinggal selama waktu tidur orang dewasa, yaitu kurang lebih 8 jam maka biaya yang akan dikeluarkan adalah sebagai berikut.

Daya saat baterai lemah = 3,68 Watt

Daya saat baterai penuh = 2,24 Watt

Harga Listrik = Rp. 1.467,28 / KWh

Waktu mengisi daya = 2 jam aktif + 6 jam dalam keadaan baterai penuh

Rp. 10,7992 + ($\frac{(2,24x6)}{1000}$x 1.467,28)

= Rp. 10,7992 + Rp. 19,7202 = Rp. 30,5194

Berdasarkan perhitungan diatas, walaupun biaya listrik yang dihasilkan sa-

ngat kecil, apabila 100 juta orang pemilik smartphone di Indonesia memiliki kebiasaan yang sama dan mengisi daya pada perangkatnya setiap hari, maka dalam jangka waktu 30 hari Indonesia akan mengeluarkan biaya listrik hanya untuk mengisi daya *smartphone* sebanyak,

$100.000.000$ orang $\times$ 30 hari $\times$ $Rp.30,5194 = Rp.91.558.200.000$ (91 miliyar rupiah).

Berikut adalah perhitungan biaya listrik apabila menggunakan SKOPIN untuk mengendalikan waktu pengisian daya

$100.000.000$ orang $\times$ 30 hari $\times$ $Rp.15,2010 = Rp.45.603.000.000$

Perbedaan selisih biaya penggunaan listrik setiap bulan pada pengisian daya *smartphone* menggunakan dan tidak menggunakan SKOPIN adalah Rp. 45.995.200.000. Masyarakat Indonesia dapat menghemat penggunaan listrik sekitar 2x lebih hemat apabila menggunakan SKOPIN. Selain menghemat biaya, menghemat penggunaan listrik dapat mencegah terjadinya pemanasan global.

2. Hasil Percobaan Batas Air pada Sensor Air

Pada sensor air, pertama kita akan mencari tahu berapa angka batas air pada arduino ketika melakukan pentetesan air yang banyak dan pencelupan sensor kedalam air setinggi 1 mm. Berikut adalah gambar *console* pada Arduino IDE ketika sensor masih dalam keadaan kering, yaitu mulai dari angka 0 dan akan naik seiring dengan jumlah air yang mengenai sensor.

Gambar 4.2: Console Arduino IDE pada saat sensor air kering

Berikut adalah gambar dari percobaan pentetesan dan pencelupan sensor ke-dalam air dan jumlah angka yang muncul pada console Arduino IDE:

1. Tetes Banyak