

ABSTRAK

Kecerobohan dalam penggunaan energi listrik, misalnya lupa mematikan lampu ketika tidak diperlukan, merupakan salah satu penyebab pemborosan energi listrik yang harus diatasi. Dampak kerugian yang akan terjadi dari sisi pengguna akibat kecerobohan tersebut salah satunya biaya yang akan terus bertambah seiring banyaknya energi yang terpakai atau lebih tepatnya terbuang. Oleh sebab itu diperlukan sebuah sistem yang mampu mengendalikan lampu-lampu tersebut agar dapat digunakan sesuai dengan kebutuhan. Pengendalian pada alat-alat listrik khususnya lampu atau penerangan merupakan hal yang penting dalam pengelolaan energi dalam suatu tempat, misalnya saja di rumah, gedung perkantoran ataupun area lainnya yang lebih luas dan mempunyai banyak lampu. Otomatisasi atau pengendalian terhadap suatu komponen elektronik ataupun listrik menjadi sangat penting di masa sekarang ini dimana keefisienan dan kecepatan dituntut dalam segala bidang agar tercapai suatu sistem yang handal serta memudahkan dalam penggunaannya. Misalnya saja pada suatu sistem pengendalian lampu pada suatu gedung atau rumah. Dengan adanya permasalahan tersebut, muncul ide atau gagasan untuk membuat sebuah sistem yang dapat mengatur penjadwalan peralatan listrik berdasarkan waktu yang telah ditentukan. Untuk memasukkan jadwal waktu menyala dan padam digunakan sebuah keypad 4x4 dan sebuah RTC yang digunakan untuk menyimpan data jam. Kedua input tersebut diproses oleh Arduino Uno R3 sebagai sistem kendalinya. Apabila jam dan menit penjadwalan sama dengan jam dan menit RTC maka lampu atau peralatan listrik yang terhubung akan menyala atau padam dikendalikan menggunakan rangkaian relay. Alat ini juga dilengkapi dengan LCD dan pengaturan tanggal dan jam secara manual apabila jam tidak sesuai dengan waktu sekarang.

Kata Kunci : *Penjadwalan Alat Listrik, RTC DS1307, Arduino Uno R3, Keypad*

ABSTRACT

Carelessness in the use of electricity, for example, forgot to turn off lights when not needed, is one cause of electrical energy waste that must be addressed. The impact of losses that will occur due to carelessness of the user side is one of them costs will continue to increase with the amount of energy used or rather wasted. It therefore requires a system capable pulled ndalikan the lights in order to digunakan according to need. Controlling the power tools especially light or illumination is essential in the management of energy in a place, such as home, office building or other area that is wider and has a lot of lights. Automation or control of an electrical or electronic component is particularly important at the present time where efficiency and speed are required in all areas in order to achieve a reliable system and ease of use. For example, in a lighting control system in a building or home. Given these problems, the idea or the idea to create a system that can manage the scheduling of electrical equipment by a predetermined time. To enter a time schedule lit and extinguished used a 4x4 keypad and an RTC that are used to store data clock. Both inputs are processed by the Arduino Uno R3 as the control system. If the scheduling of hours and minutes is equal to the hour and minute RTC then lights or electrical appliances that are connected to on or off is controlled using a relay circuit. The tool is also equipped with an LCD and setting the date and time manually if the clock does not match the current time.

Keywords: *Scheduling Electrical Equipment, RTC DS1307, Arduino Uno R3, Keypad*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kecerobohan dalam penggunaan energi listrik, misalnya lupa mematikan lampu ketika tidak diperlukan, merupakan salah satu penyebab pemborosan energi listrik yang harus diatasi. Dampak kerugian yang akan terjadi dari sisi pengguna akibat kecerobohan tersebut salah satunya biaya yang akan terus bertambah seiring banyaknya energi yang terpakai atau lebih tepatnya terbuang. Oleh sebab itu diperlukan sebuah sistem yang mampu mengendalikan lampu-lampu tersebut agar dapat digunakan sesuai dengan kebutuhan.

Pengendalian pada alat-alat listrik khususnya lampu atau penerangan merupakan hal yang penting dalam pengelolaan energi dalam suatu tempat, misalnya saja di rumah, gedung perkantoran ataupun area lainnya yang lebih luas dan mempunyai banyak lampu. Otomatisasi atau pengendalian terhadap suatu komponen elektronik ataupun listrik menjadi sangat penting di masa sekarang ini dimana keefisienan dan kecepatan dituntut dalam segala bidang agar tercapai suatu sistem yang handal serta memudahkan dalam penggunaannya. Misalnya saja pada suatu sistem pengendalian lampu pada suatu gedung atau rumah.

Namun, dewasa ini umumnya masih banyak dijumpai pengendalian saklar lampu yang dilakukan secara manual sehingga menyebabkan penggunaan alat-alat listrik tidak terkontrol dengan baik, misalnya penggunaan alat-alat yang dijalankan oleh tenaga listrik pada ruangan gedung yang terdiri dari beberapa lantai dan mempunyai banyak ruangan, akan menimbulkan masalah jika dalam pengontrolan menghidupkan dan mematikan peralatan tersebut dalam hal ini lampu listrik yang ada di setiap ruangan dilakukan secara manual.

Dari permasalahan yang ada, muncul gagasan untuk membuat suatu terobosan baru yaitu “**Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino**” yang diharapkan menjadi solusi dari permasalahan yang ada.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat diambil suatu rumusan masalah berikut.

1. Bagaimana merancang alat Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino?
2. Bagaimana cara menerapkan waktu pada jam digital sebagai acuan pada Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino?
3. Bagaimana cara mengintegrasikan antara *Arduino Uno*, *RTC (Real Time Clock)* dan *LCD* sebagai komponen utama dalam perancangan Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino ?

1.3 Batasan Masalah

Dikarenakan banyaknya cakupan permasalahan yang terdapat pada perancangan alat ini, maka penulis perlu untuk membatasi batasan masalah yaitu:

1. Menggunakan *RTC (Real Time Clock)* sebagai sistem pewaktuan (Jam Digital).
2. Menggunakan *Keypad 4x4* sebagai input password.
3. Menggunakan *Arduino Uno R3* sebagai pengolahan data input dan output sistem.
4. Menggunakan *LCD 16x2* untuk menampilkan informasi angka dan huruf.

1.4 Tujuan Penelitian

Berdasarkan perumusan masalah di atas maka tujuan yang ingin dicapai pada penulisan skripsi ini adalah sebagai berikut:

1. Untuk merancang alat Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino.
2. Untuk menerapkan waktu pada jam digital sebagai acuan pada Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino.
3. Untuk mengintegrasikan antara *Arduino Uno*, *RTC (Real Time Clock)* dan *LCD* sebagai komponen utama dalam perancangan Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino.

1.5 Manfaat Penelitian

Adapun manfaat yang diambil dalam penulisan skripsi ini adalah :

1. Dapat merancang alat Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino.
2. Dapat menerapkan waktu pada jam digital sebagai acuan pada Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino.
3. Dapat mengintegrasikan antara *Arduino Uno*, *RTC (Real Time Clock)* dan *LCD* sebagai komponen utama dalam perancangan Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino.
4. Alat serta sistem yang telah dibuat dapat diaplikasikan dalam kehidupan sehari-hari untuk digunakan dalam sistem otomasi pada rumah tangga ataupun pada kampus.

1.6 Metode Penelitian

Dalam penulisan skripsi, menggunakan beberapa metode untuk mengumpulkan data-data yang akan diperlukan untuk menyelesaikan skripsi ini.

1. Metode Studi Pustaka

Penulis melakukan studi pustaka untuk memperoleh data-data yang berhubungan dengan skripsi dari berbagai sumber bacaan seperti: buku-buku, jurnal, berkas-berkas, laporan yang berkaitan dengan judul yang diangkat sebagai referensi.

2. Metode Eksperimen

Yaitu membuat alat secara langsung dan menguji apakah Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino ini telah bekerja sesuai dengan keinginan.

3. Metode pengujian sistem

Yaitu melakukan pengujian terhadap alat atau sistem yang bertujuan untuk mengetahui apakah kinerja dari alat yang di buat sesuai dengan apa yang diharapkan atau belum.

1.7 Sistematika Penulisan

Sistematika penulisan merupakan gambaran umum dari bab isi dari penulisan skripsi.

Adapun gambaran umum dari tiap bab adalah:

BAB I : PENDAHULUAN

Pada bab ini berisi tentang pendahuluan mencakup Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan penelitian, Manfaat Penelitian, Metode Penelitian dan Sistematika Penulisan.

BAB II : LANDASAN TEORITIS

Pada bab ini berisikan tentang tinjauan pustaka relevan, teori-teori komponen yang digunakan pada Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino, seperti Arduino Uno R3, *Keypad*, *LCD*, *Buzzer* dan teori-teori pendukung lainnya.

BAB III : METODOLOGI PENELITIAN

Menganalisa komponen dan perangkat yang dibutuhkan dalam Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga menggunakan mikrokontroler Arduino berdasarkan studi literatur dan pengamatan pada objek sehingga diharapkan bisa mendapatkan hasil yang maksimal dalam perancangan sistem tersebut.

BAB IV : ANALISIS DAN PENGUJIAN

Pada bab ini membahas mengenai implementasi dari sistem yang di bangun beserta kelebihan dan kekurangan yang di peroleh.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini membahas kesimpulan yang diperoleh dari hasil pengujian serta saran-saran pengembangan yang membangun untuk kesempurnaan skripsi.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka Relevan

Perkembangan teknologi informasi baik software dan hardware menunjang perkembangan dalam berbagai bidang, khususnya dalam bidang elektronika. Kemajuan ini bisa dimanfaatkan dalam pembuatan sistem otomatisasi dan pengontrolan terhadap komponen-komponen elektronika dan listrik. Pengendalian saklar lampu listrik yang dikerjakan secara manual menyebabkan penggunaan listrik dan kontrol dalam pengoperasian lampu yang bermasalah, misalnya pengendalian lampu secara manual dalam sebuah gedung bertingkat dan memiliki banyak lampu. Hal ini menjadikan sistem manual tersebut tidak efektif dan bisa memakan banyak waktu atau menyita banyak tenaga dalam pengoperasiannya. Dengan kemajuan komputer maka dapat dibangun sebuah sistem pengendalian yang secara otomatis.

Dari hasil penelitian yang telah dilakukan oleh Iyuditya Erlina Dayanti dengan judul Sistem Pengendali Lampu Ruangan Secara Otomatis Menggunakan PC Berbasis Mikrokontroler Arduino Uno. Pengendalian lampu menggunakan komputer dengan cara menjalankan aplikasi interface menggunakan visual basic. Hasil penelitian ini sistem pengendalian lampu secara otomatis dapat meningkatkan efisiensi kerja di tempat implementasi yaitu Lab. Nepalnet SMAN 1 Palimanan dengan 42 responden. Berdasarkan tabel prosentase dapat menunjukkan bagaimana efektifitas penggunaan dalam proses pengendalian dan efisiensi, dan respon presentase adalah 78,5% dengan nilai *Cronbach's Alpha* hasil uji SPSS sebesar 0,718 makamasuk ke dalam kategori baik atau sistem ini bisa digunakan dan sangat bermanfaat serta menyempurnakan sistem yang sebelumnya telah ada.

Pada penelitian yang lain yaitu Perancangan Sistem Pengendali dan Penjadwal Lampu Ruang Berdasarkan Database Melalui Komunikasi Wireless Zigbee yang dilakukan oleh Julian Ilham, masih menggunakan interface visual basic untuk pengaturan penjadwalan lampu menyala dan padam. Aplikasi yang telah dibuat menambahkan database untuk sistem penjadwalan agar data tetap tersimpan. Hasil penelitian yang telah dilakukan menggunakan XBee-PRO yang mampu berkomunikasi sampai dengan jarak 200 meter pada saat mode line of sight, sedangkan pada mode indoor hanya sampai dengan jarak 90 meter. Pada pengujian sistem secara keseluruhan, saat mode otomatis lampu akan menyala sesuai dengan jadwal yang telah ditentukan dan banyaknya lampu yang menyala sesuai dengan jumlah mahasiswa yang tertera pada database.

2.2 Arduino

Arduino merupakan mikrokontroler yang memang dirancang untuk bisa digunakan dengan mudah oleh para seniman dan desainer. Dengan demikian, tanpa mengetahui bahasa pemrograman, Arduino bisa digunakan untuk menghasilkan karya yang canggih. Hal ini seperti yang diungkapkan oleh Mike Schmidt.

Menurut Massimo Banzi, salah satu pendiri atau pembuat Arduino, Arduino merupakan sebuah platform hardware *open source* yang mempunyai input/output (I/O) yang sederhana. (Abdul Kadir ; 2013 : 5)

Menggunakan Arduino sangatlah membantu dalam membuat suatu *prototyping* ataupun untuk melakukan pembuatan proyek. Arduino memberikan I/O yang sudah lengkap dan bisa digunakan dengan mudah. Arduino dapat digabungkan dengan modul elektro yang lain sehingga proses perakitan jauh lebih efisien.

Arduino merupakan salah satu pengembang yang banyak digunakan. Keistimewaan Arduino adalah hardware yang *Open Source*. Hal ini sangatlah memberi keleluasaan bagi

orang untuk bereksprimen secara bebas dan gratis. Secara umum, Arduino terdiri atas dua bagian utama, yaitu:

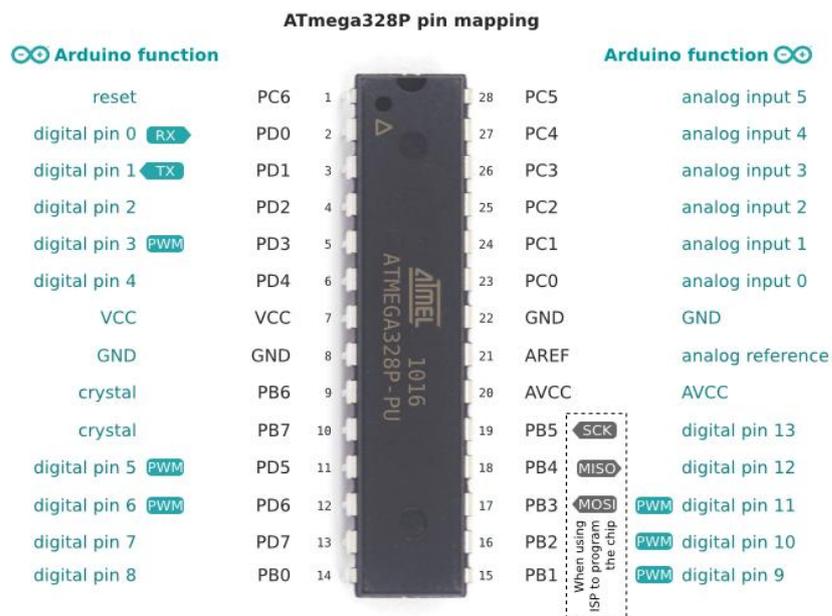
1. Bagian Hardware

Berupa sistem minimum mikrokontroler yang berisi input/output.



Gambar 2.1. Board Arduino
(Sumber: Yuwono Martha Dinata ; 2015 : 3)

Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai output PWM, 6 analog input, crystal osilator 16 MHz, koneksi USB, jack power, kepala ICSP, dan tombol reset. Arduino mampu men-support mikrokontroler; dapat dikoneksikan dengan komputer menggunakan kabel USB.



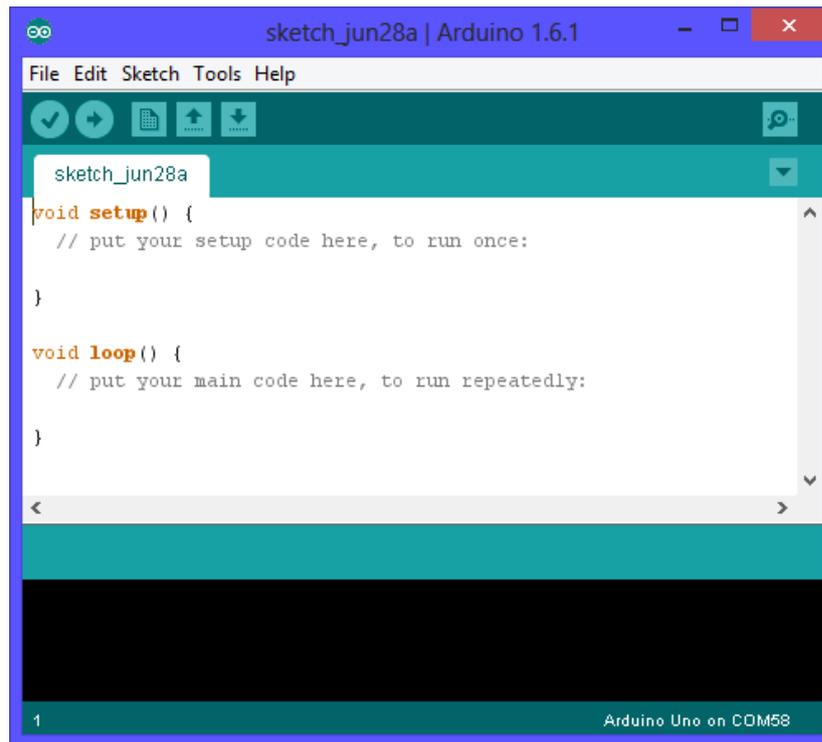
Gambar 2.2 Arduino Uno ATmega 328 Pin Mapping

Setiap 14 pin digital pada arduino dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Input/output dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maximum 40 mA dan memiliki internal pull-up resistor (disconnected oleh default) 20-50K Ohm. Beberapa pin memiliki fungsi sebagai berikut :

- Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB ke TTL chip serial.
- PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi `analogWrite()`.
- Interupt eksternal : 2 dan 3. Pin ini dapat dikonfigurasi untuk trigger sebuah interap pada low value, rising atau falling edge, atau perubahan nilai.
- SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensupport komunikasi SPI, yang mana masih mendukung hardware, yang tidak termasuk pada bahasa arduino.
- LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

2. Bagian Software

Berupa Software Arduino yang meliputi *Integrated Development Enviroment* (IDE) untuk menulis program. Arduino memerlukan instalasi driver untuk menghubungkan dengan komputer. Pada IDE terdapat contoh program dan *library* untuk pengembangan program. IDE software Arduino yang digunakan diberi nama *Sketch*, seperti Gambar 2.3.



Gambar 2.3. Software Arduino
(Sumber: Yuwono Martha Dinata ; 2015 : 4)

Contoh Penulisan *Code* Program pada Arduino Uno.

```
int i;
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
  digitalWrite(13,LOW);
  Serial.begin(9600);
  i=10;
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,LOW); delay(500);
  digitalWrite(13,HIGH); delay(500);
  Serial.print("Serial Test ");
  Serial.println(i);
  i--;
  if(i<=0) i=10;
}
```

2.3. Mikrokontroler

2.3.1. Gambaran Mikrokontroler

Tidak seperti sistem komputer, yang mampu menangani berbagai macam program aplikasi (misalnya pengolah kata, pengolah angka dan lain sebagainya), Mikrokontroler hanya bisa digunakan untuk satu aplikasi tertentu saja. Perbedaan lainnya terletak pada perbandingan RAM dan ROM-nya. Pada sistem komputer perbandingan RAM dan ROM-nya besar, artinya program-program pengguna disimpan dalam ruang RAM yang relatif besar dan rutin-rutin antarmuka perangkat keras disimpan dalam ruang ROM yang kecil. Sedangkan pada mikrokontroler, perbandingan ROM dan RAM-nya yang besar artinya program kontrol disimpan dalam ROM (bisa Masked ROM atau Flash PEROM) yang ukurannya relatif lebih besar, sedangkan RAM digunakan sebagai tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan ATMEGA328.

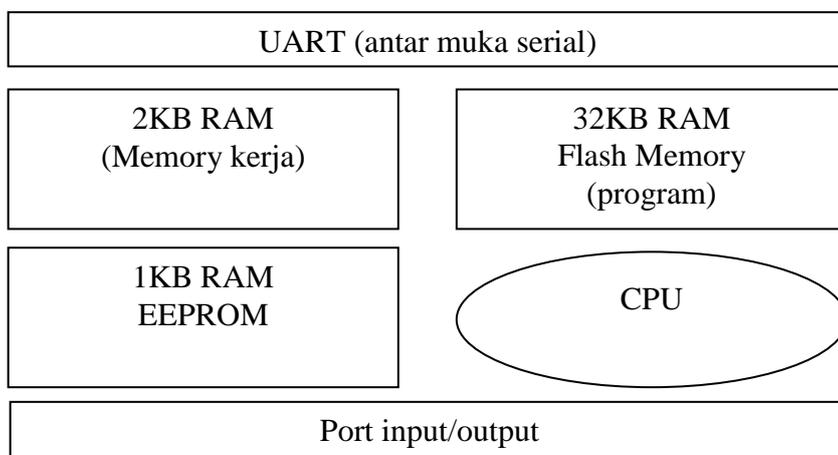
2.3.2. Mikrokontroler Arduino Uno ATmega328

Arduino Uno adalah salah satu produk berlabel arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega328 (sebuah keping yang secara fungsional bertindak seperti sebuah komputer). Peranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks.

Pengendalian LED hingga pengontrolan robot dapat di implementasikan dengan menggunakan papan yang berukuran relatif kecil ini. Arduino uno mengandung mikroprosesor (berupa atmel AVR) dan dilengkapi dengan *oscillator* 16 MHZ (yang memungkinkan operasi berbasis waktu dilaksanakan dengan tepat), dan regulator (pembangkit tegangan) 5 volt. Sejumlah pin tersedia di papan. Pin 0 hingga 13 digunakan untuk isyarat digital, yang hanya bernilai 0 atau 1. Pin A0-A5 digunakan untuk isyarat analog. Arduino Uno dilengkapi dengan *static random acces memory* (SRAM) berukuran 1 KB untuk memegang data, *flash memory* berukuran 32KB, dan *erasable programmable read-only memory* (EEPROM) untuk menyimpan perintah. (Sumber : Abdul Kadir ; 2013 : 16)

2.3.3. Arsitektur ATmega 328

Untuk memberikan gambaran mengenai apa saja yang terdapat di dalam sebuah mikrokontroler, pada gambar dibawah ini diperlihatkan contoh diagram blok sederhana dari mikrokontroler ATmega328 (dipakai pada Arduino Uno) seperti Gambar 2.4 blok diagram sederhana berikut ini:



Gambar 2.4. Arsitektur ATmega 328
(Sumber : Yuwono Martha Dinata ; 2015 : 7)

Keterangan Gambar 2.4 diatas sebagai berikut:

1. *Universal Asynchronous Receiver/Transmitter (UART)* adalah antar muka yang digunakan untuk komunikasi serial seperti pada RS-232, RS-422 dan RS-485.
2. 2KB RAM pada memory kerja bersifat *volatile* (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program.
3. 32KB RAM flash memory bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, flash memory juga menyimpan *bootloader*.

4. *Bootloader* adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah *bootloader* selesai dijalankan, berikutnya program ini akan dijalankan di dalam RAM akan dieksekusi.
5. 1KB EEPROM bersifat non-volatile, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan. Tidak digunakan pada papan Arduino.
6. *Central Processing Unit (CPU)*, bagian dari mikrokontroler untuk menjalankan setiap instruksi dari program.
7. Port input/output, pin-pin untuk menerima data (input) digital atau analog, dan mengeluarkan data (output) digital atau analog. (Sumber : Yuwono Martha Dinata ; 2015 : 7)

2.4 Software Arduino IDE

IDE (*Integrated Development Environment*) Arduino merupakan aplikasi yang mencakup *editor*, *compiler*, dan *uploader* dapat menggunakan semua seri modul keluarga arduino, seperti Arduino Duemilanove, Uno, Bluetooth, Mega. Kecuali beberapa tipe *board* produksi arduino yang memakai mikrokontroler diluar seri AVR, seperti mikroprosesor ARM. Editor sketch pada IDE arduino juga mendukung fungsi penomoran baris, mendukung fungsi penomoran baris, *syntax highlighting*, yaitu pengecekan sintaksis kode sketch. Arduino yang dipakai adalah arduino versi 1.6.4 yang terlihat pada gambar 2.5.



Gambar 2.5 Arduino IDE Versi 1.6.4

2.5. Keypad

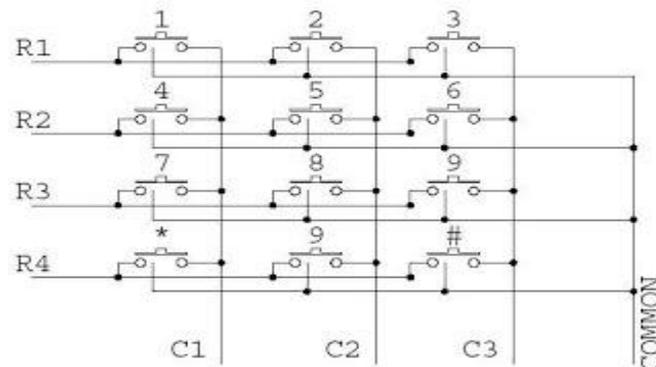
Menurut Sumardi (2013: hal 47), keypad adalah salah satu modul yang sering digunakan dalam *embedded system*. Fungsi utama modul ini adalah sebagai perangkat masukan. Secara *hardware* keypad adalah kumpulan tombol yang membentuk matriks. Dengan n baris x n kolom. Konfigurasi modul keypad dapat dilihat pada Gambar 2.6 berikut

:



Gambar 2.6. Keypad
(Sumber : Sumardi; 2013; 47)

Menurut Sumardi (2013: hal 48), teknik yang digunakan untuk mendeteksi tombol mana yang ditekan adalah dengan melakukan pengecekan baris dan kolom yang terhubung. Teknik ini sering disebut dengan scanning keypad. Kombinasi dari kemungkinan baris dan kolom yang terhubung adalah sejumlah tombol yang ditekan. Gambar modul keypad jika dilihat dari atas adalah sebagai berikut :



Gambar 2.7. Modul Keypad
(Sumber : Sumardi; 2013; 48)

2.6 Buzzer

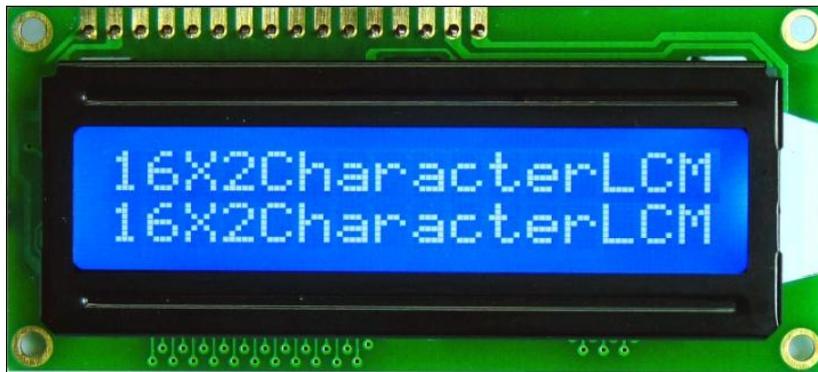
Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Buzzer berfungsi sebagai indikator suara/alarm pada perancangan alat yang akan dibuat. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).



Gambar 2.8 Bentuk Fisik Buzzer

2.7 LCD (*Liquid Crystal Display*)

Liquid Crystal Display (LCD) adalah komponen yang dapat menampilkan tulisan. Salah satu jenisnya memiliki dua baris dengan setiap baris terdiri atas enam belas karakter. LCD seperti itu biasa disebut LCD 16x2.



Gambar 2.9 LCD Karakter 16x2

LCD memiliki 16 pin dengan fungsi pin masing-masing seperti yang terlihat pada Tabel 2.1 berikut :

Tabel 2.1 Pin-pin LCD

No.Pin	Nama Pin	I/O	Keterangan
1	VSS	<i>Power</i>	Catu daya, ground (0v)
2	VDD	<i>Power</i>	Catu daya positif
3	V0	<i>Power</i>	Pengatur kontras, menurut datasheet, pin ini perlu dihubungkan dengan pin vss melalui resistor 5kΩ.
4	RS	<i>Input</i>	Register Select <ul style="list-style-type: none"> RS = HIGH : untuk mengirim data

			<ul style="list-style-type: none"> • RS = LOW : untuk mengirim instruksi
5	R/W	<i>Input</i>	Read/Write control bus <ul style="list-style-type: none"> • R/W = HIGH : mode untuk membaca data di LCD

(Sumber : Abdul Kadir ; 2013 : 196)

2.7.1 Cara kerja LCD

Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4bit atau 8bit. Jika jalur data 4 bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada table deskripsi, interface LCD merupakan sebuah parallel bus, dalam hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8bit dikirim ke LCD secara 4bit atau 8bit pada satu waktu.

Jika mode 4bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8bit (pertama dikirim 4bit MSB lalu 4bit LSB dengan pulsa clock EN setiap nibblenya). Jalur control EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus menset EN ke kondisi high “1” dan kemudian menset dua jalur control lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.

Saat jalur lainnya sudah siap, EN harus diset ke “0” dan tunggu beberapa saat, dan set EN kembali ke high “1”. Ketika jalur RS berada dalam kondisi low “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi high atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diset ke “1”. Jalur control R/W harus berada dalam kondisi low (0) saat informasi

pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi high “1”, maka program akan melakukan query data dari LCD.

Instruksi pembacaan hanya satu, yaitu Get LCD status, lainnya merupakan instruksi penulisan, Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu di set ke “0”. Jalur data dapat terdiri 4 atau 8 jalur. Mengirimkan data secara parallel baik 4bit atau 8bit merupakan 2 mode operasi primer. Untuk membuat sebuah aplikasi interface LCD, menentukan mode operasi merupakan hal yang paling penting.

Mode 8 bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3pin untuk control, 8pin untuk data). Sedangkan mode 4bit minimal hanya membutuhkan 7bit (3pin untuk control, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroler dan LCD. Jika bit ini diset ($RS = 1$), maka byte pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di reset ($RS = 0$), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca.

2.8. Relay

Menurut S Wasito (1992, h.15), *Relay* adalah suatu rangkaian *switch* magnetik yang bekerja bila mendapat catu daya. *Relay* memiliki tegangan dan arus nominal yang harus dipenuhi *output* rangkaian pen *driver* atau pengemudinya. Arus yang digunakan pada rangkaian adalah arus *DC*. [13]

Dalam dunia elektronika, relay dikenal sebagai komponen yang dapat mengimplementasikan logika switching. Sebelum tahun 70an, relay merupakan “otak” dari rangkaian pengendali. Baru setelah itu muncul PLC yang mulai menggantikan posisi relay. Relay yang paling sederhana ialah relay elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik.

Secara sederhana relay elektromekanis ini didefinisikan atau pada umum-nya Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, relay merupakan tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya.

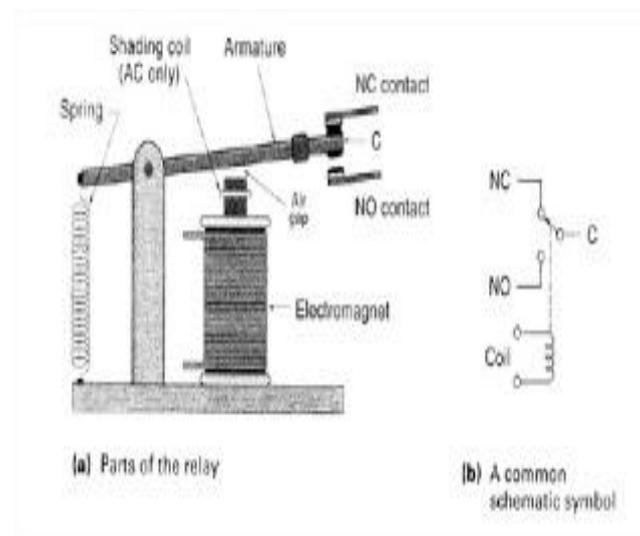
Penggunaan relay perlu memperhatikan tegangan pengontrolnya serta kekuatan relay men-switch arus/tegangan. Biasanya ukurannya tertera pada body relay. Misalnya relay 12VDC/4 A 220V, artinya tegangan yang diperlukan sebagai pengontrolnya adalah 12Volt DC dan mampu men-switch arus listrik (maksimal) sebesar 4 ampere pada tegangan 220 Volt. Sebaiknya relay difungsikan 80% saja dari kemampuan maksimalnya agar aman, lebih rendah lagi lebih aman. Relay jenis lain ada yang namanya reedswitch atau relay lidi. Relay jenis ini berupa batang kontak terbuat dari besi pada tabung kaca kecil yang dililitin kawat. Pada saat lilitan kawat dialiri arus, kontak besi tersebut akan menjadi magnet dan saling menempel sehingga menjadi saklar yang on. Ketika arus pada lilitan dihentikan medan magnet hilang dan kontak kembali terbuka (off).

Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. Relay biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 ampere AC 220 V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 ampere 12 Volt DC). Dalam pemakaiannya biasanya relay yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang di-paralel dengan lilitannya dan dipasang terbalik yaitu anoda pada tegangan (-) dan katoda pada tegangan (+).

Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat relay berganti posisi dari on ke off agar tidak merusak komponen di sekitarnya.

Secara umum, relay digunakan untuk memenuhi fungsi – fungsi berikut :

1. Remote control : dapat menyalakan atau mematikan alat dari jarak jauh
2. Penguatan daya : menguatkan arus atau tegangan Contoh : starting relay pada mesin mobil
3. Pengatur logika kontrol suatu system
4. Secara sederhana berikut ini prinsip kerja dari relay : ketika Coil mendapat energi listrik (energized), akan timbul gaya elektromagnet yang akan menarik armature yang berpegas, dan contact akan menutup. bisa dilihat pada Gambar 2.10 di bawah;



Gambar 2.10 Rangkaian relay

Selain berfungsi sebagai komponen elektronik, relay juga mempunyai fungsi sebagai pengendali sistem. Sehingga relay mempunyai 2 macam simbol yang digunakan pada Rangkaian listrik (hardware) dan Program (software). Bentuk fisik relay tampak seperti gambar 2.11 di bawah ini.



Gambar 2.11 Gambar Relay
(Sumber : <http://teknikelektronika.com>)

2.9 Bahasa C

Bahasa C adalah bahasa pemrograman yang dapat dikatakan berada antara bahasa tingkat rendah (bahasa yang berorientasi pada mesin) dan bahasa tingkat tinggi (bahasa yang berorientasi pada manusia). Seperti yang diketahui, bahasa tingkat tinggi mempunyai kompatibilitas antara platform. Karena itu, amat mudah untuk membuat program pada berbagai mesin. Berbeda halnya dengan menggunakan bahasa mesin, sebab setiap perintahnya sangat bergantung pada jenis mesin.

Pembuat bahasa C adalah Brian W. Kernighan dan Dennis M. Ritchie pada tahun 1972. C adalah bahasa pemrograman terstruktur, yang membagi program dalam bentuk blok. Tujuannya untuk memudahkan dalam pembuatan dan pengembangan program. Program yang ditulis dengan bahasa C mudah sekali dipindahkan dari satu jenis program ke bahasa program lain. Hal ini karena adanya standarisasi bahasa C yaitu berupa standar ANSI (American National Standar Institut) yang dijadikan acuan oleh para pembuat kompilernya. Pembuat bahasa C adalah Brian W. Kernighan dan Dennis M. Ritchie pada tahun 1972. C adalah bahasa pemrograman terstruktur, yang membagi program dalam bentuk blok. Tujuannya untuk memudahkan dalam pembuatan dan pengembangan program.

Program yang ditulis dengan bahasa C mudah sekali dipindahkan dari satu jenis program ke bahasa program lain. Hal ini karena adanya standarisasi bahasa C yaitu berupa standar ANSI (American National Standar Institut) yang dijadikan acuan oleh para pembuat kompilernya.

Kelebihan Bahasa C:

- Bahasa C tersedia hampir di semua jenis computer.

- Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis computer.
- Bahasa C hanya menyediakan sedikit kata-kata kunci. hanya terdapat 32 kata kunci.
- Proses *executable* program bahasa C lebih cepat
- Dukungan pustaka yang banyak.
- C adalah bahasa yang terstruktur
- Bahasa C termasuk bahasa tingkat menengah

Penempatan ini hanya menegaskan bahwa c bukan bahasa pemrograman yang berorientasi pada mesin. yang merupakan ciri bahasa tingkat rendah. melainkan berorientasi pada obyek tetapi dapat diinterpretasikan oleh mesin dengan cepat. secepat bahasa mesin. inilah salah satu kelebihan c yaitu memiliki kemudahan dalam menyusun programnya semudah bahasa tingkat tinggi namun dalam mengesekusi program secepat bahasa tingkat rendah.

Kekurangan Bahasa C:

- Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
- Bagi pemula pada umumnya akan kesulitan menggunakan pointer.

BAB III

METODOLOGI PENELITIAN

3.1 Lokasi Penelitian

Penelitian dilaksanakan di Laboratorium dasar elektronika kampus III Universitas Muhammadiyah Sumatera Utara, jalan Kapten Mukhtar Basri No.3 Glugur Darat II Medan.

3.2 Peralatan dan Bahan Penelitian

Adapun bahan dan alat yang digunakan pada penelitian ini sebagai berikut :

3.2.1 Bahan-Bahan Penelitian

Bahan-bahan yang digunakan untuk pembuatan Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino ini yaitu :

1. LCD 2 x 16 digunakan untuk menampilkan data tulisan dan Input password.
2. Keypad 4x4 digunakan sebagai input waktu penjadwalan lampu ON atau OFF.
3. Arduino Uno digunakan untuk mengontrol rangkaian keseluruhan.
4. Buzzer sebagai indikator suara.
5. Saklar ON/OFF berfungsi untuk menghubungkan dan memutuskan tegangan.
6. Timah sebagai bahan yang akan menghubungkan kaki komponen dengan jalur tembaga.
7. Kabel Jumper yang akan digunakan untuk menghubungkan jalur rangkaian yang terpisah.

8. Tiang PCB 0,5 dan 1 inchi yang akan digunakan untuk menopang PCB.
9. Triplek digunakan sebagai kotak brankas.

3.2.2 Peralatan

Peralatan penunjang yang digunakan untuk membuat Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino ini yaitu :

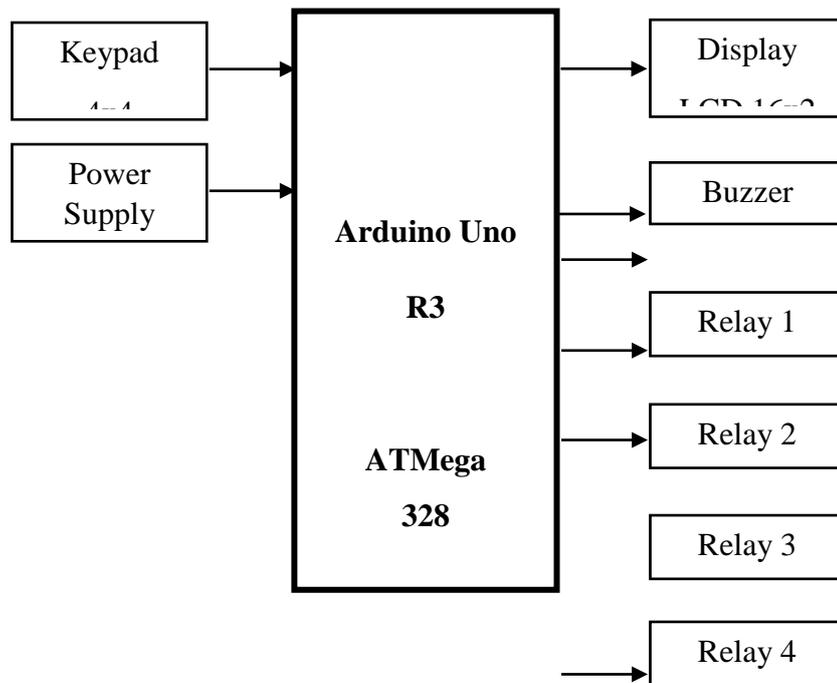
1. Power Supply 12 VDC kapasitas 1 Ampere bertujuan memberikan sumber tegangan dan Arus listrik ke perangkat Arduino.
2. Multimeter sebagai pengukur dan pengetesan komponen yang mengacu pada besaran hambatan, Arus, dan Tegangan.
3. Bor digunakan untuk membuat lubang pada PCB dan akrilik.
4. Solder untuk mencairkan timah.
5. Solder Atraktor sebagai penyedot timah.
6. Bor kayu dengan mata ukuran diameter 3 mm, dan 6 mm.
7. Penggaris untuk mengukur PCB dan Akralik.
8. Pisau Cutter untuk memotong pelat PCB dan akrilik sesuai ukuran.'
9. Tang digunakan untuk memotong maupun mengelupas kabel maupun memotong kaki komponen.

3.3. Analisa Kebutuhan

Dalam pembuatan Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino ini membutuhkan beberapa perangkat hardware dan software, antara lain:

3.3.1 Perancangan Hardware

Adapun perancangan hardware dengan menggunakan diagram blok dari sistem yang dirancang adalah seperti yang diperlihatkan pada gambar 3.1 di bawah ini.



Gambar 3.1 Diagram Blok Sistem Alat

Penjelasan dan fungsi dari masing – masing blok adalah sebagai berikut:

1. IC Mikrokontroler ATmega 328 berfungsi sebagai pusat kendali dari keseluruhan sistem kerja rangkaian.
2. Display yang digunakan adalah LCD (Liquid Crystal Display) dengan ukuran 16x2 karakter untuk menampilkan data input jadwal ON dan OFF Relay dan informasi tulisan yang lain.
3. Buzzer berfungsi sebagai indikator suara.

4. Power Supply yang digunakan berupa adaptor 12 Volt DC 1 Ampere sebagai sumber energi atau tegangan semua rangkaian *elektronika* yang telah dibuat agar bekerja sesuai perancangan.
5. Rangkaian Relay 1 – 5 sebagai saklar elektronik yang berfungsi sebagai saklar ON OFF lampu atau peralatan listrik rumah tangga lainnya.

3.3.2 Software

Software yang digunakan dalam pembuatan Rancang Bangun Implementasi Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino ini antara lain :

- 1. Proteus 8.1**

Software ini digunakan untuk menggambar skematik rangkaian.

- 2. Arduino IDE 1.6.5**

Software ini digunakan untuk penulisan program.

- 3. Ms. Office Visio**

Aplikasi software ini digunakan untuk menggambar Flowchart dari alat yang akan dibuat.

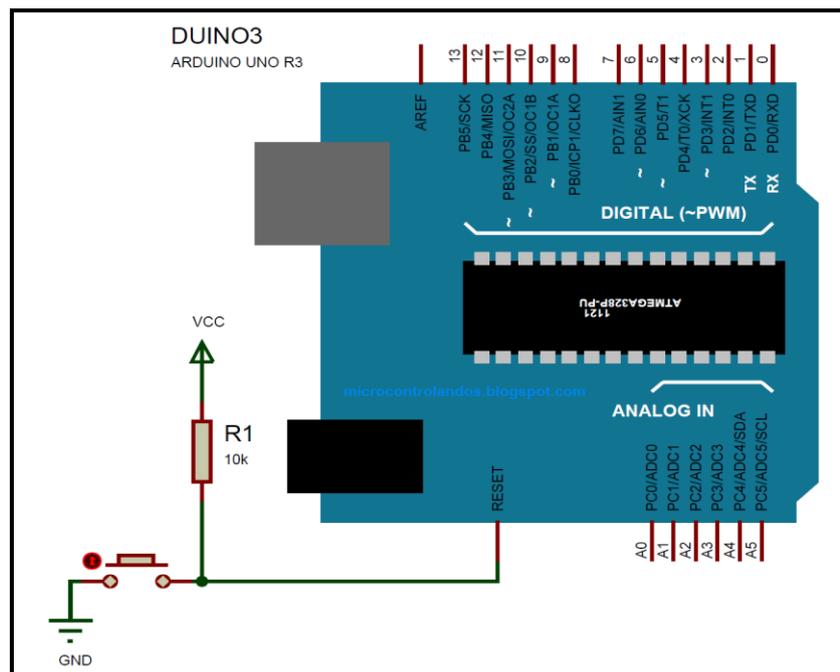
3.4. Perancangan Perangkat Keras

Pada perancangan perangkat keras ini akan dijelaskan bagaimana skematik rangkaian dari setiap blok yang sudah dijelaskan sebelumnya. Bagian-bagian perancangan perangkat keras tersebut antara lain :

3.4.1. Perancangan I/O Sistem Minimum Arduino Uno R3 ATmega328

Sistem minimum Arduino Uno R3 memiliki 14 pin I/O digital dan 6 pin I/O analog. Pin-pin tersebut dapat digunakan sebagai input seperti Keypad 4x4 dan output ke LCD karakter, Buzzer, rangkaian Relay dan lainnya.

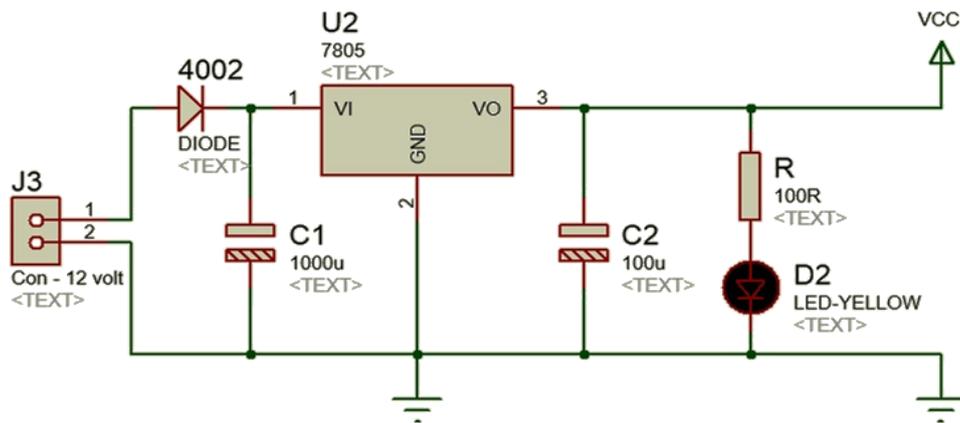
Pada Gambar 3.2. tampak jalur-jalur yang menghubungkan setiap pin I/O menuju mikrokontroler maupun jalur fitur lainnya pada sistem minimum Arduino Uno.



Gambar 3.2. Skema Rangkaian Sistem Minimum Arduino

3.4.2 . Perancangan Rangkaian *Power Supply* (PSA)

Rangkaian ini berfungsi untuk mensupply tegangan ke seluruh rangkaian yang ada. Rangkaian PSA yang dibuat terdiri dari satu keluaran, yaitu 5 volt dari input tegangan mulai dari 9 volt sampai dengan 12 volt DC. Keluaran 5 volt ini digunakan untuk *mensupply* tegangan ke semua rangkaian. Rangkaian *power supply* ditunjukkan pada gambar 3.3:

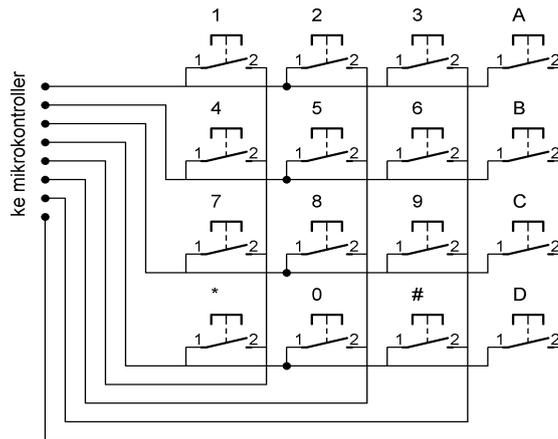


Gambar 3.3. Skematik Rangkaian *Power Supply* (PSA)

Supply tegangan berasal dari adaptor atau bisa juga menggunakan baterai yang besar tegangannya berkisar 9 volt DC sampai 12 volt DC. Kemudian tegangan tersebut akan diratakan oleh kapasitor 470 µF. Regulator tegangan 5 volt (7805) digunakan agar keluaran yang dihasilkan tetap 5 volt walaupun terjadi perubahan pada tegangan masukannya. Led hanya sebagai indikator apabila PSA dinyalakan.

3.4.3 Rangkaian Keypad

Rangkaian *Keypad* berfungsi sebagai tombol untuk memasukkan *input* jadwal ON dan OFF rangkaian relay. Rangkaian *keypad* ditunjukkan pada gambar 3.4 berikut ini :



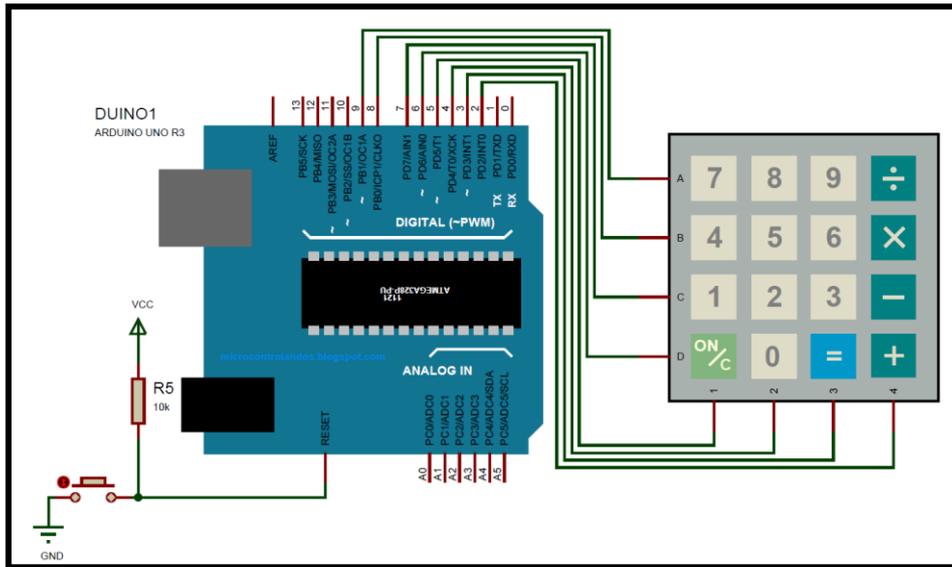
Gambar 3.4. Skematik Rangkaian *Keypad*

Rangkaian *keypad* yang digunakan adalah rangkaian *keypad* yang telah ada dipasaran. *Keypad* ini terdiri dari 16 tombol yang hubungan antara tombol-tombolnya seperti tampak pada gambar di atas. Rangkaian ini dihubungkan ke Pin 0-7 pada Arduino Uno R3.

Untuk mengakses *keypad* yang terhubung secara matriks seperti pada gambar 3.6 dapat dilakukan dengan metode *scanning*. Setiap pin baris dan pin kolom tidak terhubung satu sama lain. Saklar-saklar atau tombol tersebut jika ditekan akan menyebabkan pin baris dan pin kolom terhubung. Ketika “tombol 1” di tekan maka “tombol 1” akan menghubungkan baris 1 dengan kolom 1, ketika “tombol 3” ditekan maka “tombol 3” menghubungkan baris 1 dengan kolom 3, ketika “tombol C” ditekan maka “tombol C” menghubungkan baris 3 dengan kolom 4, dan seterusnya. Oleh sebab itu mikrokontroller harus mendeteksi terhubungnya pin baris dan pin kolom tersebut satu per satu agar tidak terjadi kesalahan pembacaan data yang selanjutnya didefinisikan sebagai suatu karakter. Untuk mencegah terjadinya kesalahan maka salah satu harus dijadikan *output* dan lainnya dijadikan *input* yaitu kolom menjadi *output* dan baris menjadi *input*.

Metode *scanning keypad* adalah mendeteksi hubungan pin baris dan kolom karena tombol ditekan, secara berurutan, bergantian dan satu-persatu. Untuk lebih jelasnya sebagai berikut :

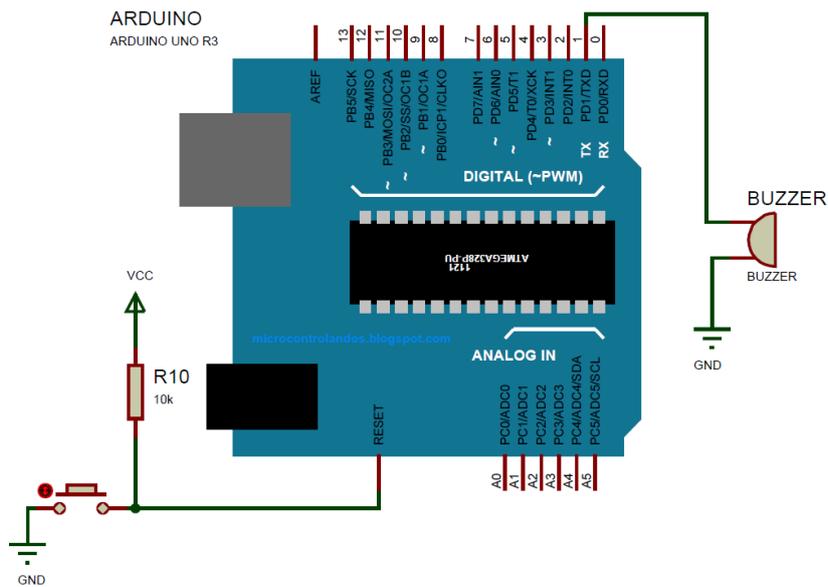
1. Pin-pin kolom menjadi output dan pin-pin baris menjadi input.
2. Pin kolom dan pin baris kondisi awalnya (default) berlogika 1 semua.
3. Output pin kolom 1 berlogika 0 lalu deteksi :
 - Jika input baris 1 berubah logika menjadi 0 maka karakter ‘1’
 - Jika input baris 2 berubah logika menjadi 0 maka karakter ‘4’
 - Jika input baris 3 berubah logika menjadi 0 maka karakter ‘7’
 - Jika input baris 4 berubah logika menjadi 0 maka karakter ‘*’
4. Pin kolom dan pin baris kembali berlogika 1 semua.
5. Output pin kolom 2 berlogika 0 lalu deteksi :
 - Jika input baris 1 berubah logika menjadi 0 maka karakter ‘2’
 - Jika input baris 2 berubah logika menjadi 0 maka karakter ‘5’
 - Jika input baris 3 berubah logika menjadi 0 maka karakter ‘8’
 - Jika input baris 4 berubah logika menjadi 0 maka karakter ‘0’
6. Pin kolom dan pin baris kembali berlogika 1 semua.
7. Output pin kolom 3 berlogika 0 lalu deteksi :
 - Jika input baris 1 berubah logika menjadi 0 maka karakter ‘3’
 - Jika input baris 2 berubah logika menjadi 0 maka karakter ‘6’
 - Jika input baris 3 berubah logika menjadi 0 maka karakter ‘9’
 - Jika input baris 4 berubah logika menjadi 0 maka karakter ‘#’
8. Pin kolom dan pin baris kembali berlogika 1 semua.
9. Output pin kolom 4 berlogika 0 lalu deteksi :
 - Jika input baris 1 berubah logika menjadi 0 maka karakter ‘A’
 - Jika input baris 2 berubah logika menjadi 0 maka karakter ‘B’
 - Jika input baris 3 berubah logika menjadi 0 maka karakter ‘C’
 - Jika input baris 4 berubah logika menjadi 0 maka karakter ‘D’



Gambar 3.5. Skematik Rangkaian Keypad dengan Arduino

3.4.4 Rangkaian Buzzer

Buzzer berfungsi sebagai indikator suara yang digunakan untuk memberikan informasi berupa suara ketika proses penekanan tombol telah dilakukan. Buzzer akan berbunyi beep-beep-beep beberapa kali. Skematik Buzzer seperti pada Gambar 3.6 berikut.

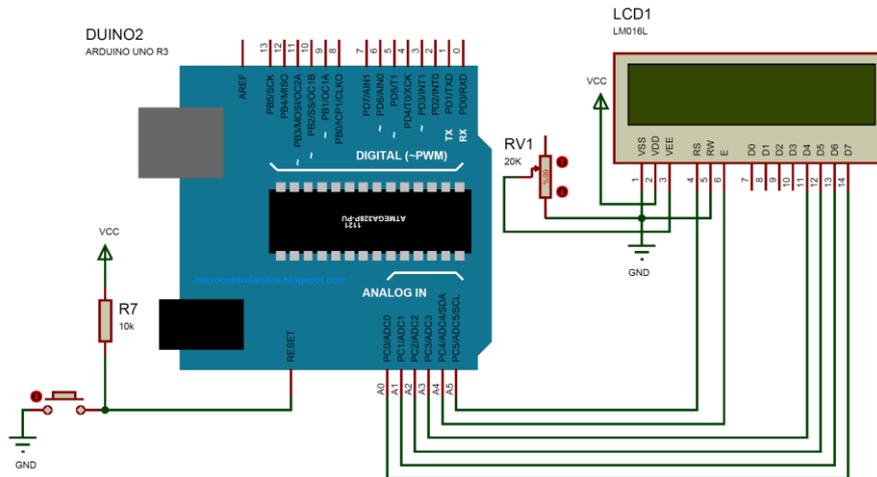


Gambar 3.6 Skematik Rangkaian Buzzer

3.4.5 Rangkaian LCD (Liquid Cristal Display)

Rangkaian LCD berfungsi untuk menampilkan kalimat dan data input jadwal.

Rangkaian LCD dapat dilihat pada Gambar 3.7 berikut ini:

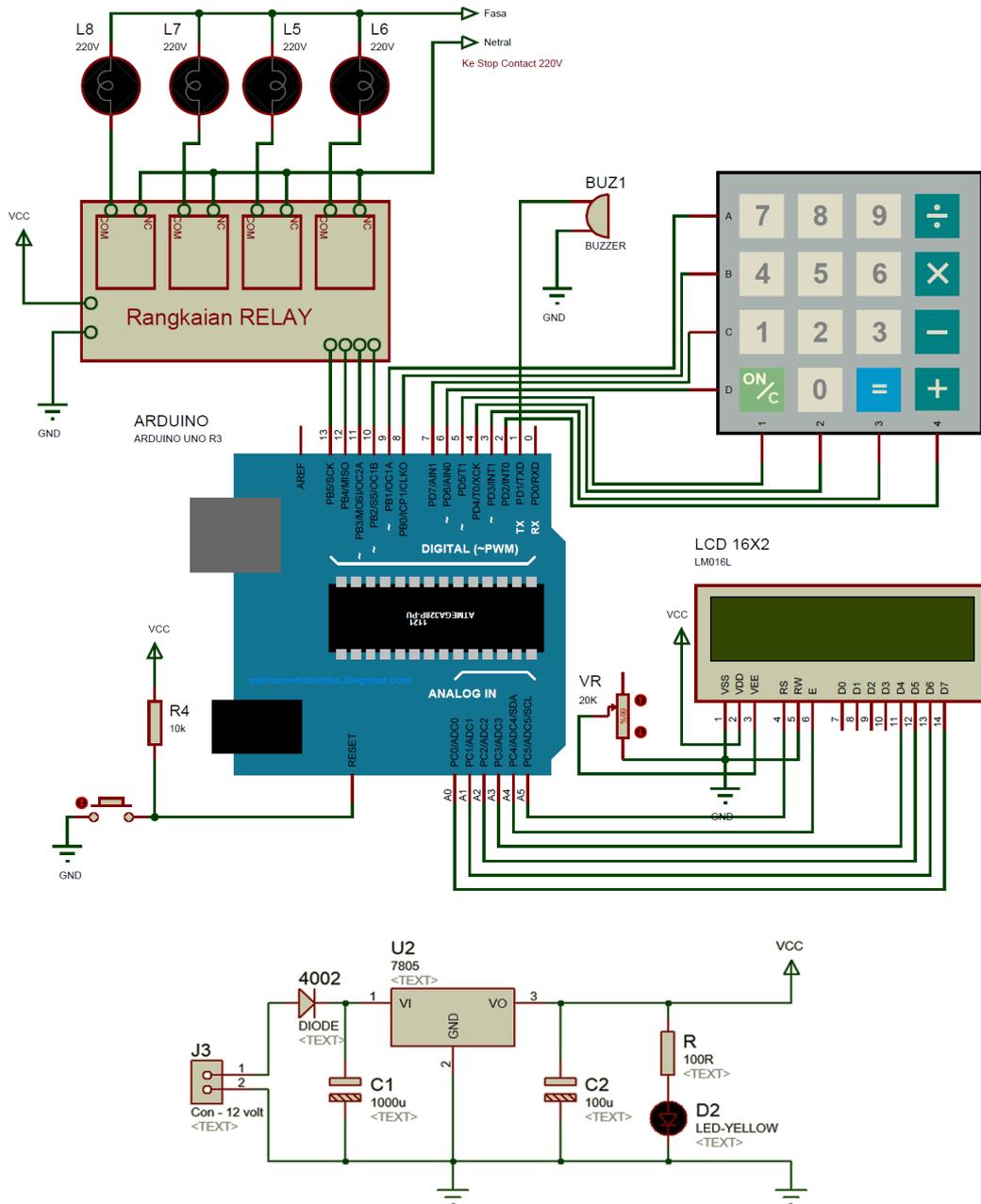


Gambar 3.7 Skematik Rangkaian LCD 16x2

Pada gambar 3.7, *pin 1* dihubungkan ke *Vcc (5V)*, *pin 2* dan *16* dihubungkan ke *Gnd (Ground)*, *pin 3* merupakan pengaturan tegangan *Contrast* dari LCD, *pin 4* merupakan *Register Select (RS)*, *pin 5* merupakan *R/W (Read/Write)*, *pin 6* merupakan *Enable*, *pin 11-14* merupakan data. *Reset*, *Enable*, *R/W* dan data dihubungkan ke *mikrokontroler ATmega328*. Fungsi dari *potensiometer (R2)* adalah untuk mengatur gelap/terangnya karakter yang ditampilkan pada LCD.

3.4.6 Rangkaian Keseluruhan

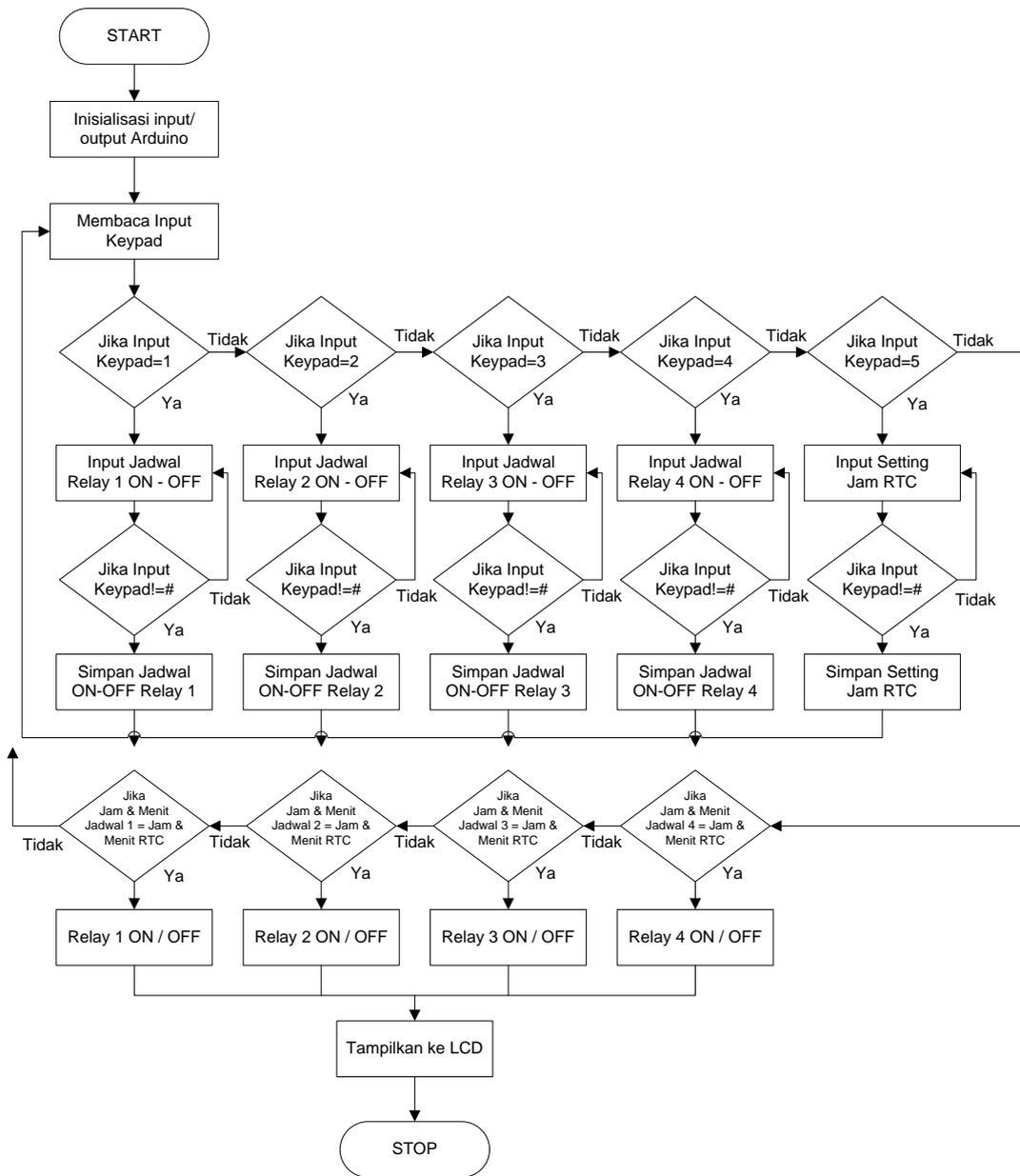
Rangkaian secara keseluruhan merupakan gabungan dari rangkaian-rangkaian tiap blok yang sudah dibahas sebelumnya. Sebagai pusat kendali Arduino Uno R3 dengan IC ATmega328 yang memproses data input RFID Reader dan input data password dari keypad. Rangkaian keseluruhan seperti Gambar 3.8 ini.



Gambar 3.8 Skematik Alat Secara Keseluruhan

Output yang digunakan yaitu buzzer sebagai indikator suara apakah sudah password sudah benar atau belum dan juga sebagai alarm apabila terjadi pembukaan pintu secara paksa. Sedangkan LCD untuk menampilkan informasi berupa tulisan dan angka seperti data input waktu jadwal ON OFF lampu atau peralatan listrik lainnya.

3.5. Flowchart



Gambar 3.9 Flowchart Sistem Kerja Alat

BAB IV

ANALISIS DAN PENGUJIAN

Dalam Bab ini akan dibahas tentang pengujian berdasarkan perencanaan dari sistem yang dibuat. Program pengujian disimulasikan di suatu sistem yang sesuai. Pengujian ini dilaksanakan untuk mengetahui kehandalan dari sistem dan untuk mengetahui apakah sudah sesuai dengan perencanaan atau belum. Pengujian pertama-tama dilakukan secara terpisah, dan kemudian dilakukan ke dalam sistem yang telah terintegrasi.

Pengujian yang dilakukan pada bab ini antara lain:

1. Pengujian Minimum Sistem Arduino Uno dengan LCD
2. Pengujian Keypad dengan LCD
3. Pengujian Rangkaian RTC
4. Pengujian Alat secara keseluruhan

4.1. Pengujian Minimum Sistem Arduino Uno dengan LCD

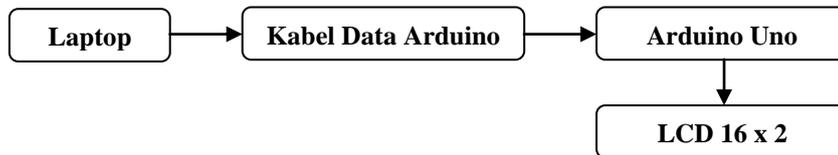
Rangkaian LCD pada penelitian ini berfungsi untuk menampilkan informasi berupa tulisan dan data dari data input RTC dan input Keypad yang dibaca oleh Arduino. Untuk mengetahui apakah rangkaian LCD yang telah dibuat dapat bekerja sesuai yang diinginkan maka dilakukan pengujian rangkaian LCD yang dihubungkan dengan minimum sistem Arduino Uno R3.

Peralatan yang dibutuhkan untuk melakukan pengujian ini yaitu :

1. Minimum Sistem Arduino Uno R3

2. Kabel data Arduino Uno R3
3. Rangkaian LCD 16 x 2
4. Software Arduino IDE

Blok diagram pengujian rangkaian LCD dengan Arduino Gambar 4.1 :



Gambar 4.1 Blok Diagram Pengujian Rangkaian LCD dengan Arduino Uno

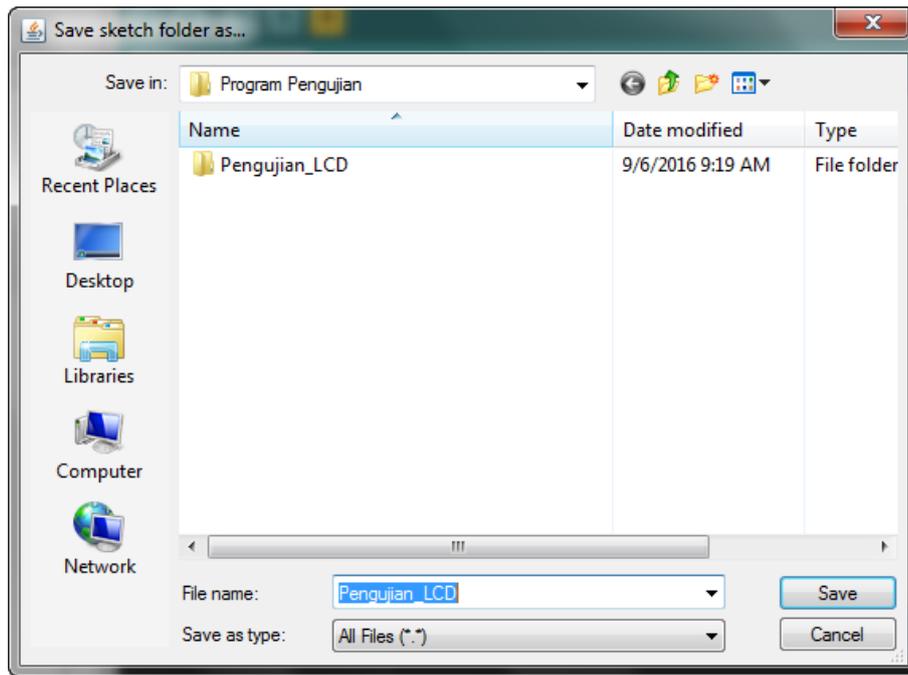
Langkah-langkah melakukan pengujian rangkaian LCD :

1. Buka aplikasi Arduino IDE 
2. Selanjutnya akan muncul tampilan awal “sketch_xxxxxx” secara otomatis seperti pada langkah sebelumnya.
3. Mengetikkan listing program untuk pengujian rangkaian LCD seperti pada gambar 4.2.

```
Pengujian_LCD $
1 #include <LiquidCrystal.h>
2 // initialize the library with the numbers of the interface pins
3 LiquidCrystal lcd(13,12,11,10,9,8);
4
5 void setup()
6 { lcd.begin(16,2);
7   lcd.setCursor(0,0); lcd.print("-LAMPU OTOMATIS-");
8   lcd.setCursor(0,1); lcd.print("BERBASIS ARDUINO");
9 }
10
11 void loop()
12 {
13
14 }
```

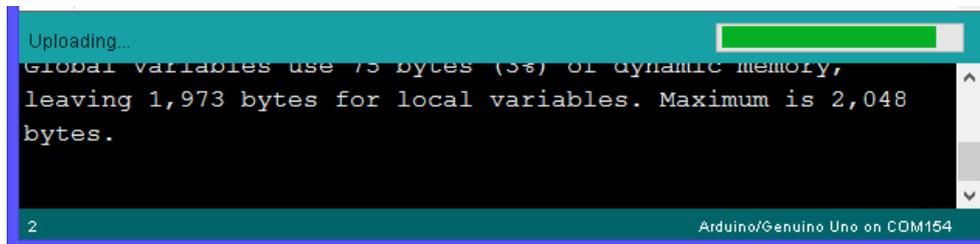
Gambar 4.2 Listing Program Pengujian LCD

4. Klik *Sketch* → *Verify*. Kemudian akan muncul kotak *dialog* untuk menyimpan *file project* yang baru dibuat. Dapat dilihat pada Gambar 4.3.



Gambar 4.3 Kotak Dialog menyimpan Program

5. Kalau sudah tidak ada *error*, maka klik ikon \rightarrow *Upload* atau *Ctrl + U*. Dapat dilihat pada gambar 4.4 di bawah ;



Gambar 4.4 Proses Uploading Program Dari Komputer Ke Arduino

Analisa Hasil Program :

Pada uji coba rangkaian *Arduino Uno* terhubung dengan LCD, diperlukan pemanggilan *library* “*LiquidCrystal lcd(13,12,11,10,9,8);*” yang berfungsi untuk menambahkan fungsi-fungsi program menampilkan karakter pada LCD. Kemudian “*lcd.begin(16,2);*” adalah *listing* program untuk pengaturan alamat LCD dan ukuran LCD jumlah baris dan kolom sesuai LCD yang digunakan. Karena yang digunakan yaitu LCD 16x2 karakter, maka *lcd_begin(16,2);*

Untuk menuliskan “-LAMPU OTOMATIS-” pada baris atas, dituliskan perintah `“lcd.setCursor(0,0); lcd.print(“-LAMPU OTOMATIS-”);`” yang artinya penulisan karakter “-LAMPU OTOMATIS-” dimulai dari kolom pertama dan baris pertama (0,0). Angka 0 menyatakan dari awal kolom dan awal baris. Apabila menginginkan penulisan pada baris kedua, yaitu menggunakan perintah `“lcd.setCursor(0,1); lcd.print (“BERBASIS ARDUINO”);` Secara keseluruhan hasil keluaran *listing program* yang ditunjukkan pada gambar 4.5



Gambar 4.5 Foto Hasil Pengujian

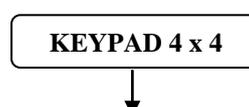
4.2 Pengujian Keypad dengan LCD

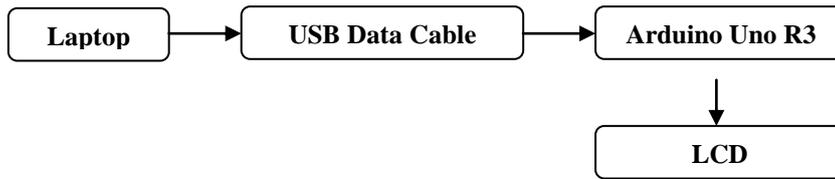
Keypad di sini merupakan papan tombol angka dan huruf yang berfungsi sebagai input data dan perintah dari pengguna ke sistem alat yang dihubungkan ke rangkaian Arduino Uno R3. Pada pengujian ini dilakukan dengan memberikan program pada mikrokontroler menampilkan data karakter berupa angka dan huruf ketika papan tombol keypad ditekan.

Peralatan yang dibutuhkan:

1. Minimum sistem Arduino Uno R3
2. Keypad 4 x 4
3. DC Power Supply
4. Seperangkat USB Data Cable
5. Software Arduino IDE

Rangkaian:





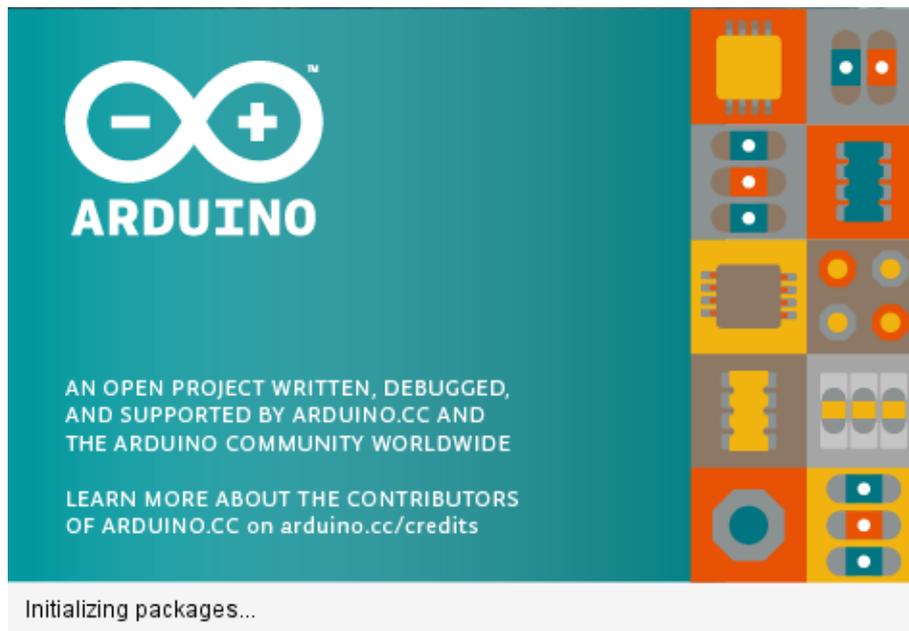
Gambar 4.6 Diagram Pengujian Keypad 4 x 4 dengan LCD

Persiapan:

1. Memasang rangkaian seperti yang ditunjukkan pada Gambar 4.6
2. Mengetik program pengujian menggunakan Software Arduino IDE.
3. Mengupload program dan Menjalankan program.

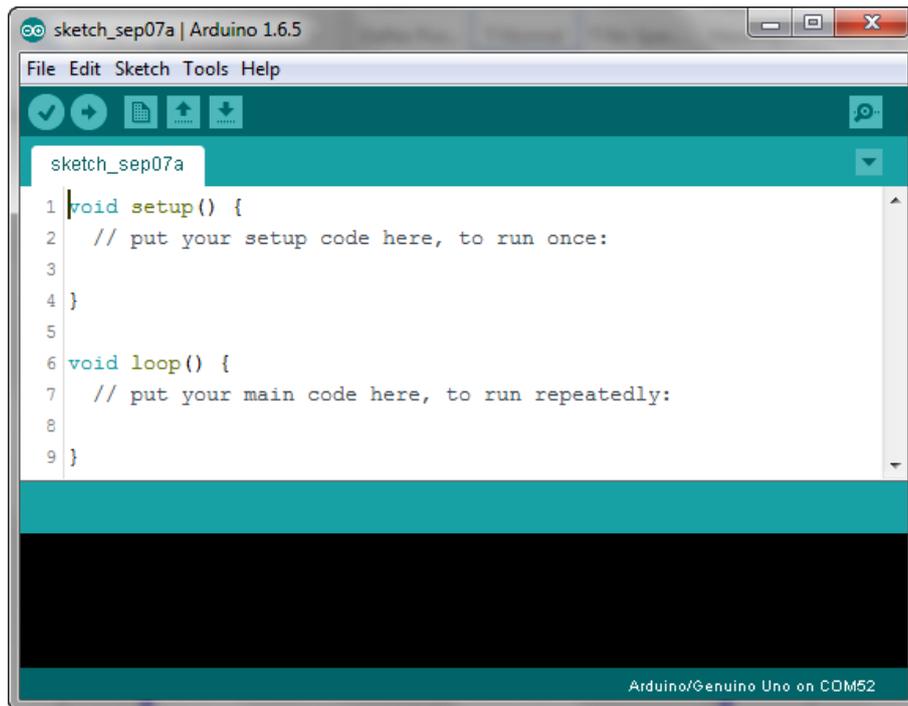
Langkah-langkah yang dilakukan:

1. Klik *Start* → *All Program* → *Arduino* → *Arduinio IDE*



Gambar 4.7 Tampilan Awal Software Arduino IDE

Selanjutnya akan muncul tampilan awal “*sketch_xxxxxx*” secara otomatis. Pada halaman inilah dimulai menuliskan program untuk melakukan pengujian rangkaian Keypad.



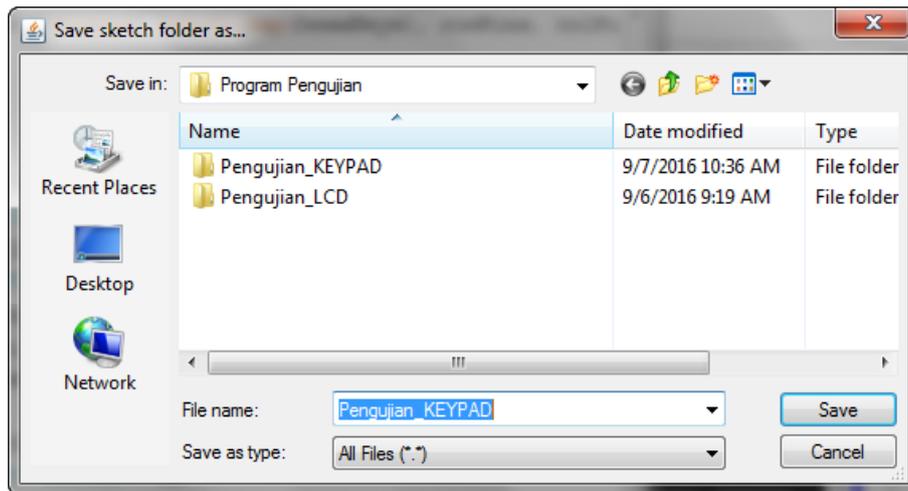
Gambar 4.8 Halaman Kerja untuk Memulai Menuliskan Program

2. Ketikkan listing program sesuai pada gambar 4.9.

```
Pengujian_KEYPAD $
1 #include <Keypad.h>
2 #include <LiquidCrystal.h>
3 // initialize the library with the numbers of the interface pins
4 LiquidCrystal lcd(13,12,11,10,9,8);
5
6 char karakter;
7 int i;
8 const byte ROWS = 4; //four rows
9 const byte COLS = 4; //four columns
10 //define the symbols on the buttons of the keypads
11 char hexaKeys[ROWS][COLS] = {
12   {'1','2','3','A'},
13   {'4','5','6','B'},
14   {'7','8','9','C'},
15   {'*','0','#','D'}
16 };
17 byte rowPins[ROWS] = {7,6,5,4}; //connect to the row pinouts of the keypad
18 byte colPins[COLS] = {3,2,1,0}; //connect to the column pinouts of the keypad
19 Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
```

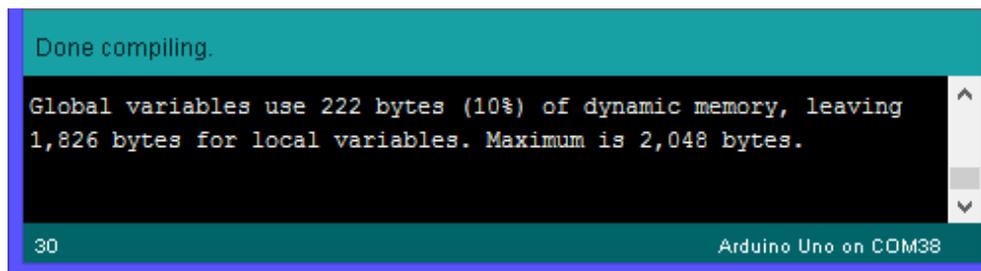
Gambar 4.9 Listing Program untuk Pengujian Test Keypad Dengan LCD

3. Kemudian Klik Sketch → Verify. Simpan dengan nama file Pengujian_KEYPAD.ino.



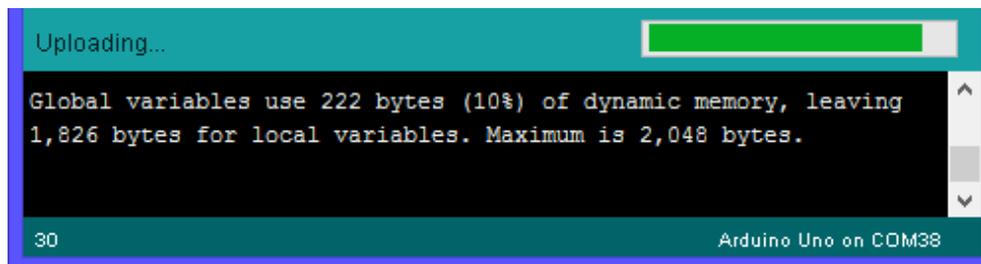
Gambar 4.10 Menyimpan Program Pengujian_KEYPAD.ino

4. Ketika sudah selesai proses Compiling, akan muncul pada bagian bawah program Arduino IDE seperti pada gambar 4.11.



Gambar 4.11 Compile Program Pengujian Keypad Dengan LCD

5. Kalau sudah tidak ada error, maka klik File → Upload atau Ctrl + U



Gambar 4.12 Upload Program ke Rangkaian Arduino Uno R3

Hasil dan Analisa :

Pada pengujian Keypad peralatan tambahan yang digunakan disini yaitu LCD yang berfungsi untuk menampilkan informasi bahwasanya tombol yang ditekan pada keypad akan ditampilkan pada LCD berupa data karakter angka dan huruf. Hasil pengujian seperti yang ditampilkan pada gambar 4.13 berikut ini.



Gambar 4.13 Tampilan LCD pada Saat Tombol Keypad Ditekan

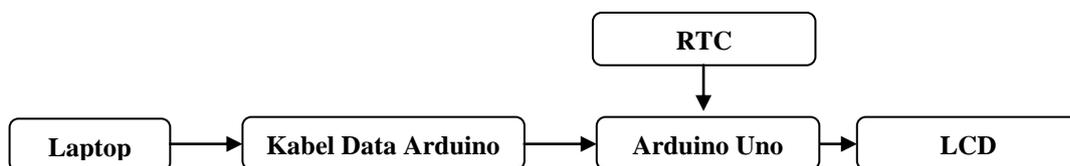
4.3 Pengujian Rangkaian RTC

Pengujian rangkaian RTC ini berfungsi untuk menyimpan data jam secara Real Time sehingga selalu tepat dengan waktu saat ini. Pengujian ini bertujuan untuk mengetahui apakah rangkaian ini dapat berfungsi dengan baik atau tidak.

Peralatan yang dibutuhkan untuk melakukan pengujian ini yaitu :

1. Minimum Sistem Arduino Uno R3
2. Rangkaian RTC
3. Software Arduino IDE

Blok diagram pengujian Rangkaian RTC seperti ditunjukkan pada Gambar 4.14 berikut ini :



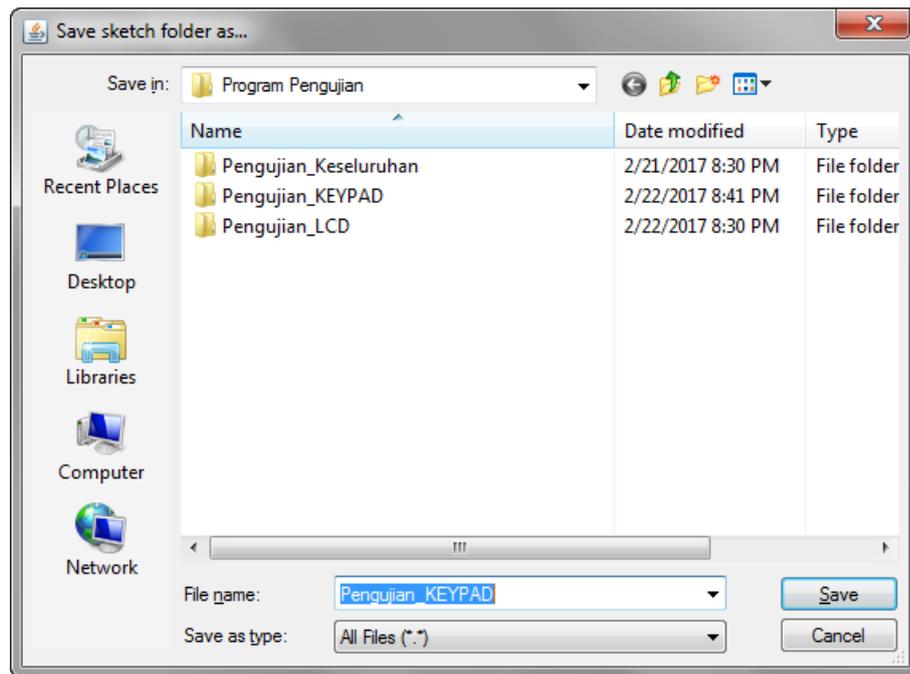
Gambar 4.14 Blok Diagram Pengujian Rangkaian RTC

Langkah-langkah melakukan pengujian Rangkaian RTC:

1. Buka aplikasi Arduino IDE

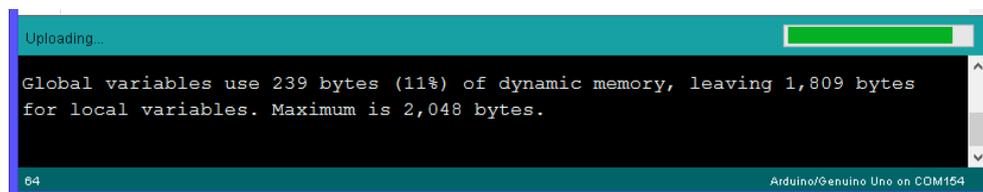


2. Selanjutnya akan muncul tampilan awal “*sketch_XXXXXX*” secara otomatis seperti pada langkah sebelumnya.
3. Mengetikkan listing program untuk pengujian Rangkaian RTC.
4. Klik *Sketch* → *Verify*. Kemudian akan muncul kotak *dialog* untuk menyimpan *file project* yang baru dibuat. Dapat dilihat pada Gambar 4.15 di bawah ini.



Gambar 4.15 Kotak Dialog menyimpan Program

5. Kalau sudah tidak ada *error*, maka klik ikon → *Upload* atau *Ctrl + U*. Dapat dilihat pada gambar 4.16 di bawah ;



Gambar 4.16 Proses Uploading Program Rangkaian RTC

Analisa Hasil Program :

Dari hasil pengujian yang telah dilakukan, untuk dapat mengetahui apakah rangkaian RTC bekerja dengan baik atau tidak dilakukan dengan menuliskan program seperti berikut ini

:

```
Pengujian_RTC $
1 #include <Wire.h> // I2C
2 #include "RTClib.h" // RTC
3 #include <stdio.h>
4 #include <LiquidCrystal.h>
5 LiquidCrystal lcd(13,12,11,10,9,8);
6 RTC_DS1307 RTC; // RTC Modul
7 char nama_hari,buffer[33];
8 int tanggal,bulan,tahun,jam,menit,detik;
9 const char* hari[] = {"Mnggu","Senin","Slasa","
10 "Rabu","Kamis","Jumat","Sabtu"};
```

Gambar 4.17 Program Pembacaan Data Jam dan Tanggal dari RTC

Gambar 4.17 merupakan sebagian listing program inisialisasi program untuk pemanggilan library yang dibutuhkan dan variabel-variabel yang dibutuhkan dalam pengujian rangkaian ini. Untuk memulai memanggil data RTC digunakan listing program berikut ini.

```
DateTime now=RTC.now();
read_datetime(now);
```

Data RTC yang telah dibaca berupa data jam dan tanggal kemudian ditampilkan pada LCD seperti ditunjukkan pada cuplikan program berikut.

```
lcd.setCursor(0,0); lcd.print(hari[nama_hari]);
sprintf(buffer,"%02d-%02d-%4d",tanggal,bulan,tahun);
lcd.print(buffer);
lcd.setCursor(0,1);
sprintf(buffer,"--%02d:%02d:%02d WIB--",jam,menit,detik);
lcd.print(buffer);
```



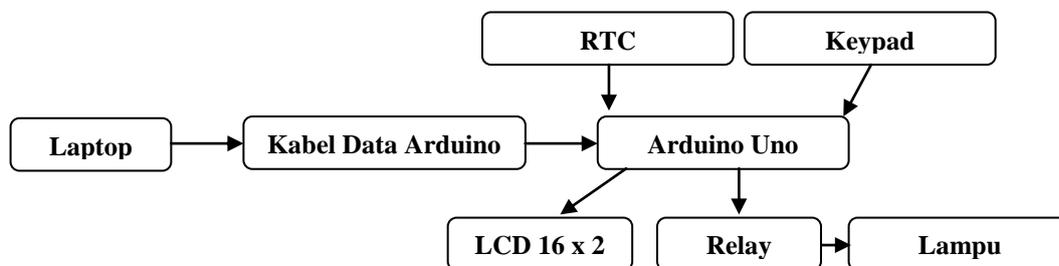
Gambar 4.18 Hasil Pembacaan Rangkaian RTC ditampilkan pada LCD

4.4. Pengujian Alat Secara Keseluruhan

Pengujian alat secara keseluruhan ini merupakan gabungan dari pengujian-pengujian tiap bagian input dan output yang telah dilakukan sebelumnya. Peralatan yang dibutuhkan untuk melakukan pengujian ini yaitu :

1. Minimum Sistem Arduino Uno R3
2. Kabel data Arduino Uno R3
3. Rangkaian Keypad
4. Rangkaian RTC
5. Rangkaian Relay
6. Rangkaian LCD
7. Software Arduino IDE

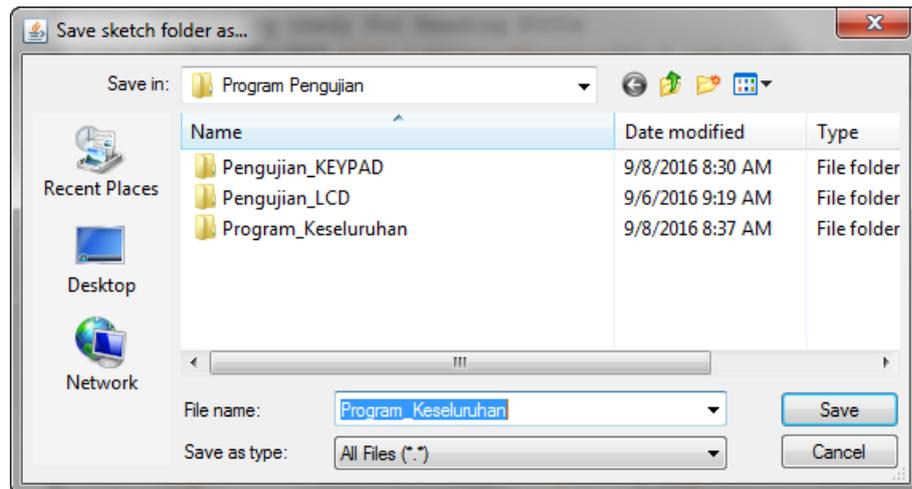
Blok diagram pengujian Alat secara Keseluruhan seperti ditunjukkan pada Gambar 4.19 berikut ini :



Gambar 4.19 Blok Diagram Pengujian Rangkaian Keseluruhan

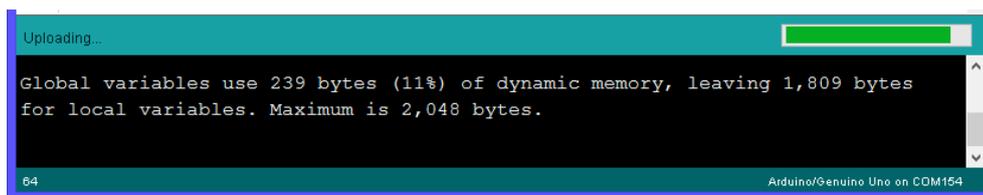
Langkah-langkah melakukan pengujian Alat secara Keseluruhan :

1. Buka aplikasi Arduino IDE 
2. Selanjutnya akan muncul tampilan awal “sketch_xxxxxx” secara otomatis seperti pada langkah sebelumnya.
3. Mengetikkan listing program untuk pengujian rangkaian Keseluruhan.
4. Klik *Sketch* → *Verify*. Kemudian akan muncul kotak *dialog* untuk menyimpan *file project* yang baru dibuat. Dapat dilihat pada Gambar 4.20 di bawah ini.



Gambar 4.20 Kotak Dialog menyimpan Program

5. Kalau sudah tidak ada *error*, maka klik ikon → *Upload* atau *Ctrl + U*. Dapat dilihat pada gambar 4.21 di bawah ;



Gambar 4.21 Proses Uploading Program Dari Komputer Ke Arduino

Analisa Hasil Program :

Proses awal, yaitu pengenalan pin dan pemanggilan library yang dibutuhkan oleh arduino. Library tersebut mencakup LCD, Komunikasi SPI, Keypad, dan RTC.

```

1 #include <Wire.h> // I2C
2 #include "RTClib.h" // RTC
3 #include <stdio.h>
4 #include <Keypad.h>
5 #include <LiquidCrystal.h>
6 // initialize the library with the numbers of the interface pins
7 LiquidCrystal lcd(13,12,11,10,9,8);
8 RTC_DS1307 RTC; // RTC Modul
9
10 unsigned int suhu,i,j,digit,nilai_jam_nyala,nilai_minit_nyala,nilai_jam_padam,nilai_minit_padam;
11 unsigned int jam_lampu1_nyala,jam_lampu1_padam,menit_lampu1_nyala,menit_lampu1_padam;
12 unsigned int jam_lampu2_nyala,jam_lampu2_padam,menit_lampu2_nyala,menit_lampu2_padam;
13 unsigned int jam_lampu3_nyala,jam_lampu3_padam,menit_lampu3_nyala,menit_lampu3_padam;
14 unsigned int jam_lampu4_nyala,jam_lampu4_padam,menit_lampu4_nyala,menit_lampu4_padam;
15 char update_jam[10];

```

Gambar 4.22 Program Pengenalan Variabel dan Library

Sebelum program utama dijalankan, perlu dilakukan inisialisasi input dan output yang digunakan tiap pin arduino. Inisialisasi tersebut berada di dalam fungsi void setup() { }.

```
40 //initialize an instance of class NewKeypad
41 Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
42
43 void setup()
44 { lcd.begin(16,2);
45   Wire.begin();
46   RTC.begin();
47   pinMode(lampu_1,OUTPUT); digitalWrite(lampu_1,1);
48   pinMode(lampu_2,OUTPUT); digitalWrite(lampu_2,1);
49   pinMode(lampu_3,OUTPUT); digitalWrite(lampu_3,1);
50   pinMode(lampu_4,OUTPUT); digitalWrite(lampu_4,1);
51
52   if (!RTC.isrunning()) {lcd.setCursor(0,0); lcd.print("RTC Error...."); }
53   else {lcd.setCursor(0,0); lcd.print("RTC Okeeee...."); }
54   //RTC.adjust(DateTime(__DATE__, __TIME__));
55   //RTC.adjust(DateTime(2016, 6, 12, 14, 52, 0));
56   delay(1000);
57   lcd.setCursor(0,0); lcd.print("-LAMPU OTOMATIS-");
58   lcd.setCursor(0,1); lcd.print("BERBASIS ARDUINO");
59   delay(4000);
60   lcd.clear();
61 }
```

Gambar 4.23 Program Inisialisasi Input dan Output

Pada cuplikan program di atas, dijelaskan LCD, Komunikasi SPI, Pembacaan Rangkaian RTC dan Keypad diinisialisasi, dan program ini hanya dijalankan 1 kali. Pada bagian `RTC.adjust(DateTime(__DATE__, __TIME__));` yaitu untuk mengatur jam pada RTC sesuai dengan tanggal dan jam pada komputer. Pada bagian ini cukup diaktifkan satu kali saja untuk mengatur tanggal dan jam pada RTC.

```

233 void read_datetime(DateTime datetime) {
234     nama_hari=datetime.dayOfWeek();
235     tanggal=datetime.day();
236     bulan=datetime.month();
237     tahun=datetime.year();
238
239     jam=datetime.hour();
240     menit=datetime.minute();
241     detik=datetime.second();
242 }
243 void tampil_datetime() {
244     lcd.setCursor(0,0);
245     lcd.print(hari[nama_hari]);
246     sprintf(buffer,"%02d-%02d-%4d",tanggal,bulan,tahun);
247     lcd.print(buffer);
248
249     lcd.setCursor(0,1);
250     sprintf(buffer,"--%02d:%02d:%02d WIB--",jam,menit,detik);
251     lcd.print(buffer);
252 }

```

Gambar 4.24 Program Utama Pembacaan Rangkaian RTC

Untuk mendapatkan data jam pada RTC, maka dibutuhkan variable untuk menampung data tanggal dan jam sementara kemudian ditampilkan pada LCD. Variable yang dibutuhkan yaitu tanggal, bulan, tahun, jam, menit dan detik seperti ditunjukkan pada gambar 4.24. variable-variable inilah yang dijadikan sebagai acuan data jam dan menit pada penjadwalan 4 buah lampu / peralatan listrik.

Pada pengujian alat secara keseluruhan ini, penulisan program disesuaikan dengan flowchart yang telah dibuat. Alur program dari proses kerja alat ini yaitu ketika Keypad angka 1 ditekan, maka akan masuk menu pengaturan lampu jam menyala sampai dengan jam padam yang diinginkan.

```

if(karakter=='1') {
    kesini1: i=0;
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("SETTING LAMPU: 1");
    lcd.setCursor(0,1); lcd.print("JAM:      ");
    while(karakter!='#') {
        karakter = customKeypad.getKey();
        if(karakter) {
            if(karakter=='*') { goto kesini1; }
            if(karakter!='A' && karakter!='B' && karakter!='C' &&
                karakter!='*') {
                lcd.setCursor(i+5,1);
                if(i==2 || i==8) lcd.print(":");
                else if(i==5) lcd.print("-");
                else {

```

```

        lcd.print(karakter);
        freq[i]=karakter;
    }
    i++;
}
}
} //end
konversi();
EEPROM.write(jam_lampul_nyala,nilai_jam_nyala);
EEPROM.write(menit_lampul_nyala,nilai_menit_nyala);
EEPROM.write(jam_lampul_padam,nilai_jam_padam);
EEPROM.write(menit_lampul_padam,nilai_menit_padam);
lcd.clear();
lcd.print("TERSIMPAN...");
digitalWrite(lampu_1,HIGH);
delay(1000);
}

```

Pada cuplikan program tersebut untuk menjalankan perintah input keypad ketika ditekan tombol 1



Gambar 4.25 Tampilan Ketika Setting Lampu 1

Pada program setting lampu di atas, menunjukkan pada pengaturan lampu 1 dijadwalkan menyala pada jam 20:22:00 dan padam pada jam 20:25:00. Cuplikan program untuk proses pengecekan penjadwalan lampu 1.

```

tampil_datetime();

if(jam==EEPROM.read(jam_lampul_nyala) && jam!=0 && menit!=0 &&
    menit==EEPROM.read(menit_lampul_nyala)) digitalWrite(lampu_1,LOW);

if(jam==EEPROM.read(jam_lampul_padam) && jam!=0 && menit!=0 &&
    menit==EEPROM.read(menit_lampul_padam)) digitalWrite(lampu_1,HIGH);

```

Apabila data jam dan menit pada RTC sama dengan jam_lampul_nyala dan menit_lampul_nyala maka relay 1 diaktifkan sehingga lampu 1 menyala.



Gambar 4.26 Tampilan Ketika Setting Lampu 2

Pada program setting lampu di atas, menunjukkan pada pengaturan lampu 2 dijadwalkan menyala pada jam 20:25:00 dan padam pada jam 20:27:00. Cuplikan program untuk proses pengecekan penjadwalan lampu 2.

```
if(jam==EEPROM.read(jam_lampu2_nyala) && jam!=0 && menit!=0 &&
    menit==EEPROM.read(menit_lampu2_nyala)) digitalWrite(lampu_2,LOW);
if(jam==EEPROM.read(jam_lampu2_padam) && jam!=0 && menit!=0 &&
    menit==EEPROM.read(menit_lampu2_padam)) digitalWrite(lampu_2,HIGH);
```

Apabila data jam dan menit pada RTC sama dengan jam_lampu2_nyala dan menit_lampu2_nyala maka relay 2 diaktifkan sehingga lampu 2 menyala.



Gambar 4.27 Tampilan Ketika Setting Lampu 3

Pada program setting lampu di atas, menunjukkan pada pengaturan lampu 3 dijadwalkan menyala pada jam 20:27:00 dan padam pada jam 20:29:00. Cuplikan program untuk proses pengecekan penjadwalan lampu 3.

```

if(jam==EEPROM.read(jam_lampu3_nyala) && jam!=0 && menit!=0 &&
  menit==EEPROM.read(menit_lampu3_nyala)) digitalWrite(lampu_3,LOW);
if(jam==EEPROM.read(jam_lampu3_padam) && jam!=0 && menit!=0 &&
  menit==EEPROM.read(menit_lampu3_padam)) digitalWrite(lampu_3,HIGH);

```

Apabila data jam dan menit pada RTC sama dengan jam_lampu3_nyala dan menit_lampu3_nyala maka relay 3 diaktifkan sehingga lampu 3 menyala.



Gambar 4.28 Tampilan Ketika Setting Lampu 4

Pada program setting lampu di atas, menunjukkan pada pengaturan lampu 4 dijadwalkan menyala pada jam 20:22:00 dan padam pada jam 20:25:00. Cuplikan program untuk proses pengecekan penjadwalan lampu 4.

```

if(jam==EEPROM.read(jam_lampu4_nyala) && jam!=0 && menit!=0 &&
  menit==EEPROM.read(menit_lampu4_nyala)) digitalWrite(lampu_4,LOW);
if(jam==EEPROM.read(jam_lampu4_padam) && jam!=0 && menit!=0 &&
  menit==EEPROM.read(menit_lampu4_padam)) digitalWrite(lampu_4,HIGH);

```

Apabila data jam dan menit pada RTC sama dengan jam_lampu4_nyala dan menit_lampu4_nyala maka relay 4 diaktifkan sehingga lampu 4 menyala.

Pada keempat proses gambar di atas, menunjukkan pengaturan waktu untuk menyalakan dan mematikan lampu 1 sampai dengan lampu 4. Waktu menyala dan waktu padam akan dibandingkan dengan data jam pada RTC. Jika sesuai maka lampu akan dinyalakan atau dimatikan melalui rangkaian relay.



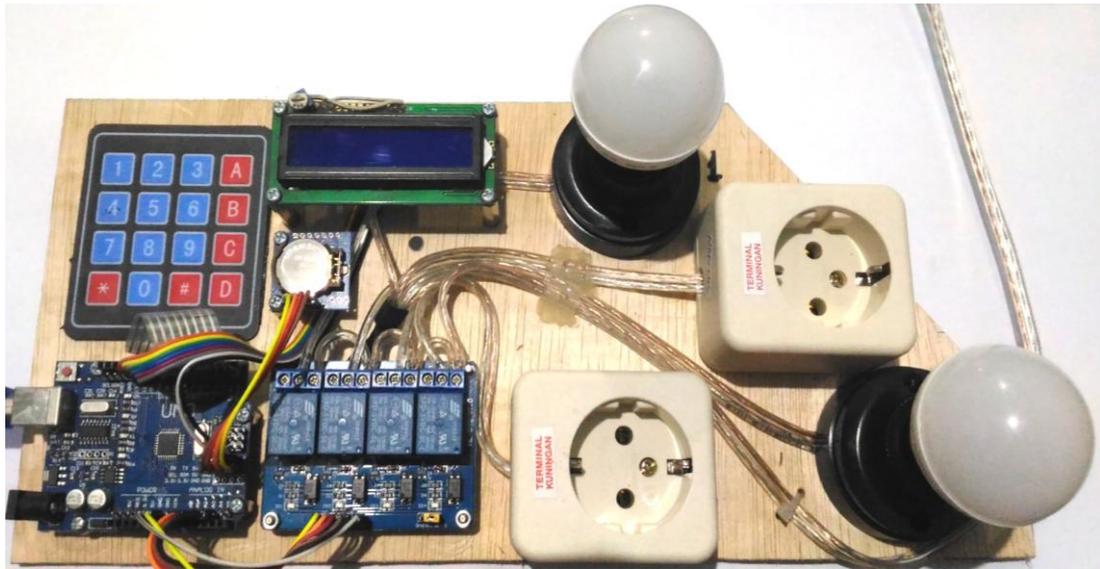
Gambar 4.29 Tampilan Ketika Setting RTC

Untuk mengatasi masalah apabila terjadi kesalahan jam pada alat, maka ditambahkan proses untuk setting jam/waktu pada alat agar sesuai dengan waktu pada saat ini, sehingga alat ini selalu update.



Gambar 4.30 Tampilan LCD Setiap Proses Berhasil Dijalankan

Pada setiap proses, yaitu proses pengaturan jam menyala dan padam pada lampu 1 sampai dengan lampu 4 dan juga pengaturan RTC, pada LCD ditampilkan tulisan Tersimpan.



Gambar 4.31 Foto Alat Secara Keseluruhan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pembuatan alat Penjadwalan Peralatan Listrik Rumah Tangga Menggunakan Mikrokontroler Arduino dan kemudian dilakukan pengujian dan analisisnya sehingga didapatkan kesimpulan sebagai berikut :

1. Alat yang telah dibuat menggunakan Arduino Uno R3 sebagai sistem kendali utamanya untuk membaca data input dari RTC (*Real Time Clock*) dan kemudian ditampilkan pada LCD karakter 16x2. Selain itu menggunakan 4 buah relay yang digunakan sebagai saklar lampu dan *stop contact*.
2. Penerapan waktu pada alat ini menggunakan rangkaian RTC DS1307 dan sebuah battery CR2032 sebagai sumber tegangan cadangan RTC agar jam pada alat selalu *up to date*. Untuk membaca data penjadwalan yang telah dibuat, Apabila data jadwal jam dan menit yang telah dibuat dengan menekan keypad tombol 1-4 sama dengan jam dan menit pada RTC maka Relay akan aktif / non aktif.
3. Rangkaian Arduino dihubungkan dengan Keypad dan RTC (*Real Time Clock*) menggunakan kabel sebagai data masukan yang diproses oleh Arduino Uno R3. Sedangkan untuk mengendalikan peralatan listrik yang telah dipersiapkan menggunakan rangkaian relay 4 chanel yang dapat mengatur ON dan OFF 2 buah lampu dan 2 stop kontak yang bisa dihubungkan dengan peralatan listrik lain. Untuk mengetahui data jam dan tanggal ditambahkan rangkaian LCD untuk menampilkan jam, tanggal dan input data dari keypad.

5.2 Saran

Untuk pengembangan lebih lanjut dari alat ini agar lebih sempurna, maka diberikan beberapa saran sebagai berikut :

1. Karya ini masih bisa dikembangkan lebih lanjut, disempurnakan dan juga adanya penambahan-penambahan lainnya, seperti penambahan sistem sms gateway yang dapat mengendalikan rangkaian relay secara langsung menggunakan perintah SMS dari Telepon Selular pemilik.
2. Sistem penjadwalan dapat disempurnakan dengan membuat aplikasi pada smartphone android yang terhubung dengan arduino.
3. RTC yang digunakan masih memiliki selisih waktu dengan jam pada umumnya, jadi penulis memiliki saran untuk mengganti modul RTC dengan yang lebih akurat agar sesuai dengan jam pada umumnya.

DAFTAR PUSTAKA

- Abdul Kadir, "*From Zero to a Pro*", Penerbit Andi, Yogyakarta, 2014.
- Abdul Kadir, "*Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya menggunakan Arduino*", Penerbit Andi, Yogyakarta, 2013.
- Heri Andrianto, Aan Darmawan, "*Belajar Cepat dan Pemrograman Arduino*", Penerbit Informatika, Bandung, 2015.
- Iyuditya, Erlina Dayanti, "*Sistem Pengendali Lampu Ruangan Secara Otomatis Menggunakan PC Berbasis Mikrokontroler Arduino Uno*", Jurnal Online ICT STMIK IKMI – Vol.10-Edisi Desember, Cirebon, 2013.
- Jazi Eko Istiyanto, "*Pengantar Elektronika dan Instrumentasi*", Penerbit Andi, Yogyakarta, 2014.
- Julian Ilham, "*Perancangan Sistem Pengendali Dan Penjadwal Lampu Ruangan Berdasarkan Database Melalui Komunikasi Wireless Zigbee*", Teknik Elektro Universitas Diponegoro, Semarang, 2012.
- Muhammad Faizun, "*Pemrograman Mikrokontroller ATmega dengan CAVR dan Simulasi ISIS Proteus*", Penerbit Deepublish, Yogyakarta, 2012.
- Syahban Rangkuti, "*Mikrokontroler Atmel AVR, Simulasi dan Praktik Menggunakan ISIS Proteus dan CodeVisionAVR*", Penerbit Informatika Bandung, Bandung, 2011.

