



HEART & MIND TOWARDS EXCELLENCE

**RANCANG BANGUN TIMER UNTUK STOP KONTAK AC  
MENGUNAKAN ARDUINO**

**TUGAS AKHIR**



**INSTITUT BISNIS  
DAN INFORMATIKA**

**stikom**

SURABAYA

Oleh :

**AKHMAD RIZKI FAJAR  
11410200038**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

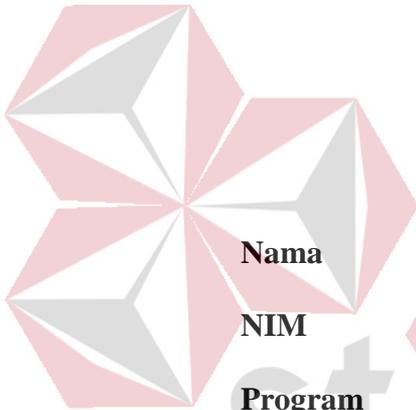
**2019**

**RANCANG BANGUN TIMER UNTUK STOP KONTAK AC  
MENGUNAKAN ARDUINO**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan**

**Program Sarjana Komputer**



Oleh :

**Nama : Akhmad Rizki Fajar**

**NIM : 11410200038**

**Program : S1 (Strata Satu)**

**Jurusan : Sistem Komputer**

INSTITUT BISNIS  
DAN INFORMATIKA  
stikom  
SURABAYA

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2019**

*Lihat dari apa yang ia perbuat,*

*Jangan hanya lihat dari luarnya saja*

*Karena itu hanya tipuan sesaat yang bisa mengalihkan segalanya.*

**Akhmad Rizki Fajar**



INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA

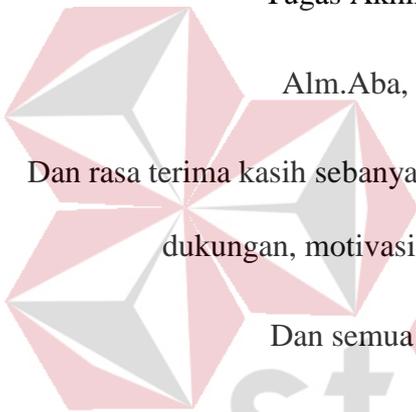
Alhamdulillah akhirnya penulis dapat menyelesaikan Tugas Akhir ini dengan baik

Tugas Akhir ini saya persembahkan kepada

Alm.Aba, Umik dan Kakak saya tercinta

Dan rasa terima kasih sebanyak mungkin kepada orang-orang yang memberikan dukungan, motivasi serta do'a yang dikhususkan bagi saya.

Dan semua orang yang sangat menyayangi saya.



INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA

**TUGAS AKHIR**  
**RANCANG BANGUN TIMER UNTUK STOP KONTAK AC**  
**MENGGUNAKAN ARDUINO**

Dipersiapkan dan disusun oleh

**Akhmad Rizki Fajar**

**NIM : 11410200038**

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji  
Pada : Februari 2019

Susunan Dewan Penguji

**Pembimbing**

**Pauladie Susanto, S.Kom., M.T.**

**NIDN. 0729047501**

**Ira Puspasari, S.Si., M.T.**

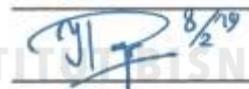
**NIDN. 0710078601**

**Penguji**

**Heri Pratikno, M.T., MTCNA., MTCRE.**

**NIDN. 0716117302**

 08/02/2019

 8/2/19



Tugas Akhir ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana



**Dr. Jusak**  
**Dekan Fakultas Teknologi dan Informatika**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**  
**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

## PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Akhmad Rizki Fajar  
NIM : 11410200038  
Program Studi : SI Sistem Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **Rancang Bangun Timer Untuk Stop Kontak AC Menggunakan Arduino**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Februari 2019  
Yang menyatakan,



Akhmad Rizki Fajar  
Nim : 11410200038

## ABSTRAK

Peralatan elektronik merupakan kebutuhan manusia yang menjadi kebutuhan pokok pada masa sekarang. Peralatan elektronik semakin diminati oleh banyak masyarakat karena dapat membantu berbagai macam pekerjaan menjadi lebih mudah, efektif serta efisien. Seperti contohnya pada pengguna *smartphone*. Data Kementerian Perindustrian (Kemenperin) mencatat, saat ini terdapat 24 perusahaan manufaktur komponen produk ponsel dan tablet di dalam negeri. Sementara itu, berdasarkan laporan e-Marketer, pengguna aktif *smartphone* di Indonesia akan tumbuh dari 55 juta orang pada tahun 2015 menjadi 100 juta orang tahun 2018.

Namun semakin berkembang dan banyaknya pengguna peralatan elektronik, semakin banyak pula kelalaian dalam penggunaannya. Sebagai contoh *overcharging* pada *gadget*, televisi yang masih menyala meskipun sudah tidak di lihat.

Sebagian besar orang tidak mengawasi peralatan elektroniknya khususnya *smartphone* yang sedang di-charge kemudian ditinggal untuk melakukan aktifitas lain ataupun ditinggal tidur semalaman yang akan membuat perangkat tersebut mengalami *Overcharging*. *Overcharging* adalah kondisi dimana proses 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.

Dari beberapa kasus yang terjadi dalam masyarakat di atas, maka dibuatlah suatu alat yang dapat membantu masyarakat untuk mengurangi resiko.

Telah dilakukan penelitian untuk melakukan pengendalian dan penjadwalan arus listrik. Penelitian yang telah dilakukan menghasilkan dua mode penjadwalan. Pada mode pertama, pengujian telah dilakukan pada stop kontak 1 dengan memberi beberapa nilai waktu 1 menit, 3 menit dan 5 menit dan diuji pada 3 objek yang berbeda dengan tingkat keberhasilan 100%. Pada stop kontak 2 dengan memberi beberapa nilai waktu 2 menit, 4 menit dan 6 menit dan diuji pada 3 objek yang berbeda dengan tingkat keberhasilan 100%. Pada stop kontak 3 dengan memberi beberapa nilai waktu 2 menit, 5 menit dan 7 menit dan diuji pada 3 objek yang berbeda dengan tingkat keberhasilan 100%.

Pada mode kedua, pengujian telah dilakukan pada stop kontak 1 dengan memberi beberapa nilai waktu 1 menit, 3 menit dan 5 menit dan diuji pada 3 objek yang berbeda dengan tingkat keberhasilan 100%. Pada stop kontak 2 dengan memberi beberapa nilai waktu 2 menit, 4 menit dan 6 menit dan diuji pada 3 objek yang berbeda dengan tingkat keberhasilan 100%. Pada stop kontak 3 dengan memberi beberapa nilai waktu 2 menit, 5 menit dan 7 menit dan diuji pada 3 objek yang berbeda dengan tingkat keberhasilan 100%.

Kata Kunci : Arduino, stop kontak, AC, *timer*, *Relay*

## KATA PENGANTAR

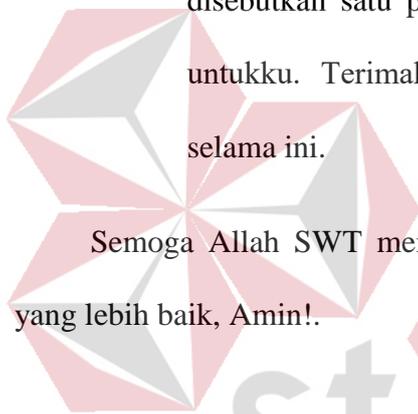
Puji syukur tetap terlimpahkan pada Allah SWT yang telah memberikan rahmat dan hidayah-Nya penulis berhasil menyelesaikan penulisan Tugas Akhir ini dengan sebaik-baiknya. Penulis mengambil judul “RANCANG BANGUN TIMER UNTUK STOP KONTAK AC MENGGUNAKAN ARDUINO” ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Institut Bisnis dan Informatika STIKOM Surabaya.

Pada kesempatan kali ini penulis juga ingin mengucapkan terima kasih kepada:

1. Orang Tua dan Saudari saya yang telah memberikan banyak dukungan, motivasi dan doa.
2. PimPinan STIKOM Surabaya yang telah banyak memberikan motivasi serta teladan yang dapat membantu penulis selama menempuh pembelajaran hingga saat ini.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Kaprodi S1 Sistem Komputer serta dosen pembimbing pertama yang telah membantu serta mendukung setiap kegiatan sehingga pelaksanaan Tugas Akhir ini dapat berjalan dengan baik.
4. Ibu Ira Puspasari, S.Si., M.T. selaku dosen wali serta dosen pembimbing kedua penulis yang senantiasa memberikan dukungan kepada penulis sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.

5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE. selaku pembahas yang telah membimbing penulis yang memberi masukan dalam menyusun buku Tugas Akhir dan membimbing selama menempuh perkuliahan di Institut Bisnis dan Informatika STIKOM Surabaya.
6. Seluruh dosen Pengajar Program Studi S1 Sistem Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Institut Bisnis dan Informatika STIKOM Surabaya.
7. Teman-teman satu jurusan dan semua pihak yang tidak bisa disebutkan satu per satu. Kalian semua selalu jadi yang istimewa untukku. Terimakasih buat bantuan, dukungan dan do'a kalian selama ini.

Semoga Allah SWT membalas seluruh kebaikan kalian dengan kebaikan yang lebih baik, Amin!.



INSTITUT BISNIS  
DAN INFORMATIKA  
**stikom**  
SURABAYA

Surabaya, Februari 2019

Penulis

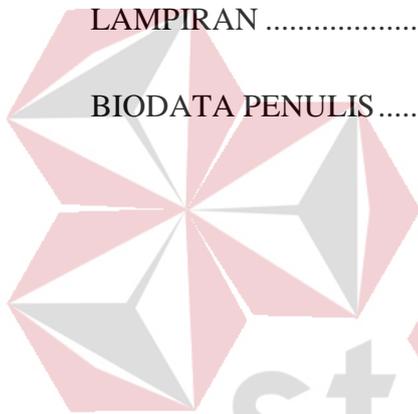
## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN SYARAT .....	ii
MOTTO .....	iii
HALAMAN PERSEMBAHAN .....	iv
HALAMAN PENGESAHAN .....	v
HALAMAN PERNYATAAN .....	<b>Error! Bookmark not defined.</b>
ABSTRAK .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xviii
BAB I .....	1
PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan .....	4
1.4 Batasan Masalah .....	5
1.5 Sistematika Penulisan .....	5
BAB II .....	7

LANDASAN TEORI .....	7
2.1  Arduino Mega 2560.....	7
2.2  LCD OLED .....	12
2.3 <i>Switch</i> .....	12
2.4  RTC ( <i>Real Time Clock</i> ) DS3231.....	14
2.5 <i>Relay</i> .....	16
2.6  Stop Kontak.....	18
<b>BAB III</b> .....	20
<b>METODE PENELITIAN DAN PERANCANGAN SISTEM</b> .....	20
3.1  Metode Penelitian.....	20
3.2  Model Perancangan .....	20
3.3  Perancangan Sistem.....	22
3.4  Rencana Menu Yang Akan di Tampilkan Pada Layar LCD OLED .....	23
3.5  Perancangan Perangkat Keras .....	25
3.5.1  Perancangan RTC .....	27
3.5.2  Perancangan LCD OLED .....	27
3.5.3  Perancangan <i>Relay</i> .....	28
3.5.4  Perancangan <i>Push Button</i> .....	29
3.6  Perancangan Perangkat Lunak .....	29
3.6.1 <i>Flowchart</i> Mode Pertama .....	30

3.6.2	Flowchart Mode Kedua.....	31
3.6.3	Flowchart Keseluruhan Sistem.....	32
3.7	Metode Analisis.....	33
BAB IV.....		34
hasil pengujian dan pengamatan.....		34
4.1.	Pengujian Arduino.....	34
4.1.1	Tujuan.....	34
4.1.2	Alat Yang Di Gunakan.....	34
4.1.3	Prosedur Pengujian.....	35
4.1.4	Hasil Pengujian.....	35
4.2.	Pengujian LCD OLED.....	37
4.2.1	Tujuan.....	37
4.2.2	Alat Yang Digunakan.....	37
4.2.3	Prosedur Pengujian.....	37
4.2.4	Hasil Pengujian.....	38
4.3.	Pengujian RTC ( <i>Real Time Clock</i> ).....	40
4.3.1	Tujuan.....	40
4.3.2	Alat yang digunakan.....	40
4.3.3	Prosedur Pengujian.....	40
4.3.4	Hasil Pengujian.....	44
4.4.	Pengujian Sistem.....	45

4.4.1 Pengujian Mode Pertama.....	45
4.4.2 Pengujian Mode Kedua.....	54
4.4.3 Penjelasan Hasil Uji Coba .....	63
BAB V .....	67
5.1 Kesimpulan.....	67
5.2 Saran.....	67
DAFTAR PUSTAKA.....	68
LAMPIRAN .....	69
BIODATA PENULIS.....	79



INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Arduino Mega 2560 Sisi Depan (Kiri) dan Belakang (Kanan)	8
<b>Gambar 2.2</b> LCD OLED .....	12
<b>Gambar 2.3</b> <i>Switch</i> .....	13
<b>Gambar 2.4</b> Modul RTC .....	15
<b>Gambar 2.5</b> Gambar <i>Relay</i> dan Simbol <i>Relay</i> .....	16
<b>Gambar 2.6</b> Cara Kerja <i>Relay</i> .....	17
<b>Gambar 2.7</b> Gambar Stop Kontak.....	19
<b>Gambar 3.1</b> Gambar Perancangan Alat.....	20
<b>Gambar 3.2</b> Blok Diagram Sistem .....	22
<b>Gambar 3.3</b> Gambar Rancangan Menu.....	23
<b>Gambar 3.4</b> Gambar Rancangan Perangkat Keras.....	25
<b>Gambar 3.5</b> Gambar Rancangan Arduino Dengan RTC.....	27
<b>Gambar 3.6</b> Gambar Rancangan Arduino Dengan LCD OLED.....	28
<b>Gambar 3.7</b> Gambar Rancangan Arduino Dengan <i>Relay</i> .....	28
<b>Gambar 3.8</b> Gambar Rancangan Arduino Dengan Push Button.....	29
<b>Gambar 3.9</b> <i>Flowchart</i> Mode Menu Pertama .....	30
<b>Gambar 3.10</b> <i>Flowchart</i> Mode Menu Kedua .....	31
<b>Gambar 3.11</b> <i>Flowchart</i> Keseluruhan Sistem .....	32
<b>Gambar 4.1</b> <i>upload</i> program berhasil .....	36
<b>Gambar 4.2</b> Program berhasil berjalan .....	36
<b>Gambar 4.3</b> <i>upload</i> program berhasil .....	39
<b>Gambar 4.4</b> Program berhasil berjalan .....	39

<b>Gambar 4.5</b> <i>upload</i> program berhasil .....	44
<b>Gambar 4.6</b> Program Berhasil Berjalan .....	45
<b>Gambar 4.7</b> Posisi Pertama Alat di Nyalakan.....	45
<b>Gambar 4.8</b> Hasil Percobaan Setelah Satu Menit .....	46
<b>Gambar 4.9</b> Posisi Pertama Alat di Nyalakan.....	46
<b>Gambar 4.10</b> Hasil Percobaan Setelah Tiga Menit.....	47
<b>Gambar 4.11</b> Posisi Pertama Alat di Nyalakan.....	47
<b>Gambar 4.12</b> Hasil Percobaan Setelah Lima Menit.....	48
<b>Gambar 4.13</b> Posisi Pertama Alat di Nyalakan.....	48
<b>Gambar 4.14</b> Hasil Percobaan Setelah Satu Menit .....	49
<b>Gambar 4.15</b> Posisi Pertama Alat di Nyalakan.....	49
<b>Gambar 4.16</b> Hasil Percobaan Setelah Tiga Menit.....	50
<b>Gambar 4.17</b> Posisi Pertama Alat di Nyalakan.....	50
<b>Gambar 4.18</b> Hasil Percobaan Setelah Lima Menit.....	51
<b>Gambar 4.19</b> Posisi Pertama Alat di Nyalakan.....	51
<b>Gambar 4.20</b> Hasil Percobaan Setelah Satu Menit .....	52
<b>Gambar 4.21</b> Posisi Pertama Alat di Nyalakan.....	52
<b>Gambar 4.22</b> Hasil Percobaan Setelah Tiga Menit.....	53
<b>Gambar 4.23</b> Posisi Pertama Alat di Nyalakan.....	53
<b>Gambar 4.24</b> Hasil Percobaan Setelah Lima Menit.....	54
<b>Gambar 4.25</b> Posisi Pertama Alat di Nyalakan.....	54
<b>Gambar 4.26</b> Hasil Percobaan Setelah Satu Menit .....	55
<b>Gambar 4.27</b> Posisi Pertama Alat di Nyalakan.....	55
<b>Gambar 4.28</b> Hasil Percobaan Setelah Tiga Menit.....	56

<b>Gambar 4.29</b> Posisi Pertama Alat di Nyalakan.....	56
<b>Gambar 4.30</b> Hasil Percobaan Setelah Lima Menit.....	57
<b>Gambar 4.31</b> Posisi Pertama Alat di Nyalakan.....	57
<b>Gambar 4.32</b> Hasil Percobaan Setelah Satu Menit .....	58
<b>Gambar 4.33</b> Posisi Pertama Alat di Nyalakan.....	58
<b>Gambar 4.34</b> Hasil Percobaan Setelah Dua Menit.....	59
<b>Gambar 4.35</b> Posisi Pertama Alat di Nyalakan.....	59
<b>Gambar 4.36</b> Hasil Percobaan Setelah Tiga Menit.....	60
<b>Gambar 4.37</b> Posisi Pertama Alat di Nyalakan.....	60
<b>Gambar 4.38</b> Hasil Percobaan Setelah Satu Menit .....	61
<b>Gambar 4.39</b> Posisi Pertama Alat di Nyalakan.....	61
<b>Gambar 4.40</b> Hasil Percobaan Setelah Tiga Menit.....	62
<b>Gambar 4.41</b> Posisi Pertama Alat di Nyalakan.....	62
<b>Gambar 4.42</b> Hasil Percobaan Setelah Lima Menit.....	63



## DAFTAR TABEL

<b>Tabel 2.1</b> Spesifikasi Arduino Mega 2560 .....	8
<b>Tabel 3.1</b> Tabel Hubungan <i>Pin</i> Modul Dengan <i>Pin</i> Arduino .....	26
<b>Tabel 4.1</b> Tabel Percobaan Saklar 1 Mode 1 .....	64
<b>Tabel 4.2</b> Tabel Percobaan Saklar 2 Mode 1 .....	64
<b>Tabel 4.3</b> Tabel Percobaan Saklar 3 Mode 1 .....	64
<b>Tabel 4.4</b> Tabel Percobaan Saklar 1 Mode 2 .....	65
<b>Tabel 4.5</b> Tabel Percobaan Saklar 2 Mode 2 .....	65
<b>Tabel 4.6</b> Tabel Percobaan Saklar 3 Mode 2 .....	66



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Peralatan elektronik merupakan kebutuhan manusia yang menjadi kebutuhan pokok pada masa sekarang. Seiring dengan terus berkembang dan banyaknya jenis peralatan elektronik yang semakin bertambah, semakin banyak pula minat dari pengguna yang ingin memilikinya. Peralatan elektronik semakin diminati oleh banyak masyarakat karena dapat membantu berbagai macam pekerjaan menjadi lebih mudah, efektif serta efisien. Seperti contohnya pada pengguna *smartphone*. Data Kementerian Perindustrian (Kemenperin) mencatat, saat ini terdapat 24 perusahaan manufaktur komponen produk telepon genggam dan tablet di dalam negeri. Sementara itu, berdasarkan laporan e-Marketer, pengguna aktif *smartphone* di Indonesia akan tumbuh dari 55 juta orang pada tahun 2015 menjadi 100 juta orang tahun 2018 (Novalius, 2018).

Namun semakin berkembang dan banyaknya pengguna peralatan elektronik, semakin banyak pula kelalaian dalam penggunaannya. Sebagai contoh *overcharging* pada *gadget*, televisi yang masih menyala meskipun sudah tidak di lihat.

Sebagian besar orang tidak mengawasi peralatan elektroniknya khususnya *smartphone* yang sedang di-charge kemudian ditinggal untuk melakukan aktifitas lain ataupun ditinggal tidur semalaman yang akan membuat perangkat tersebut mengalami *Overcharging*. *Overcharging* adalah kondisi dimana proses 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 (Prastiantari, 2017). Hal ini dapat menyebabkan kerusakan pada perangkat elektronik, dalam hal ini *smartphone*. Kelalaian pada *smartphone* juga terjadi ketika tetap meninggalkan steker menetap pada stop kontak. Apabila steker berada pada stop kontak terlalu lama dan terjadi penumpukan steker pada stop kontak, hal ini dapat menyebabkan terjadinya kerusakan pada kabel-kabel dan komponen penghantar listrik karena menerima arus yang terlalu besar.

Televisi juga termasuk dalam salah satu peralatan yang luput dari pengawasan, seperti seringnya membiarkan televisi menyala tetapi tidak dilihat karena berbagai alasan. Misalnya ketiduran atau melakukan pekerjaan lain. Hal ini dapat menyebabkan pemborosan penggunaan listrik. Peralatan listrik seharusnya menyala hanya jika dibutuhkan saja dan terbatas pada waktu yang di butuhkan. Misalnya, *charger gadget* hanya beroperasi selama 2 jam, televisi yang hanya menyala selama 4 jam atau sesuai kebutuhan yang di inginkan, dan masih banyak lagi.

Selain permasalahan di atas, penjadwalan pengaliran arus listrik juga diperlukan. Sebagai contoh yang terjadi pada bel sekolah. Dimana bel sekolah sekarang masih di nyalakan dan di matikan secara manual menggunakan tenaga manusia. Hal ini dirasa kurang efisien karena terkadang manusia bisa lalai.

Adakalanya, manusia sering lupa untuk mematikan saklar lampu rumahnya karena sibuk dengan urusan masing-masing, seperti bekerja, sekolah, ataupun pergi keluar kota. Terkadang karena terlalu sibuk, mereka tidak sempat untuk

mematikan lampu ruangnya, walaupun hari sudah siang lampu ruangnya masih menyala. Hal tersebut tentunya akan berakibat pada melonjaknya biaya tagihan listrik, dan pemborosan energi yang bisa merugikan pengguna rumah (Fatimah, 2017). Misalnya ketika pemilik rumah meninggalkan rumah dalam jangka waktu lama, sehingga memungkinkan timbul suatu tindakan kriminal, seperti pencurian, karena dianggap di rumah tersebut sedang kosong.

Alat yang sudah ada dan sudah diperjual belikan pada masyarakat ialah suatu timer yang hanya dapat mengatur satu alat saja dan hanya terdapat satu pengaturan saja. Sehingga hal ini tidak memungkinkan jika kita ingin mengontrol perangkat elektronik lebih dari satu dalam satu stop kontak AC.

Ada beberapa penelitian yang pernah dilakukan oleh Dini Destiani Siti Fatimah dengan judul “Perancangan Pengendali Lampu Rumah Otomatis Berbasis Arduino Nano” dan Alitinia Prastiantari dengan judul “SKOPIN (STOP KONTAK PINTAR) PENGENDALI ARUS LISTRIK MENGGUNAKAN TIMER PADA STOP KONTAK BERBASIS ARDUINO” dimana keduanya masih mengontrol hanya satu mode dan jumlah stop kontak yang diatur masih satu buah.

Dari beberapa kasus yang terjadi dalam masyarakat diatas, maka dibuatlah suatu alat yang dapat membantu masyarakat untuk mengurangi resiko. Alat ini dibuat dengan 2 menu. Menu pertama adalah memberikan pengatur waktu sebagai pengendali *OFF* untuk aliran listrik yang mengalir stop kontak ketika alat sudah selesai digunakan. Menu kedua adalah membuat penjadwalan secara otomatis untuk mengendalikan arus listrik yang mengalir stop kontak, sehingga alat yang di gunakan dapat menyala dan mati secara teratur sesuai keinginan.

Pengendalian dan penjadwalan arus listrik ini menggunakan Arduino sebagai penghubung antara pengatur waktu (*timer*) dan *Relay* yang tujuannya untuk mengatur durasi dan menentukan waktu yang digunakan untuk mengalirkan atau menghentikan aliran listrik pada setiap perangkat. Alat ini akan berfungsi sebagai pengendali listrik, sehingga aliran listrik akan tersambung dan terputus sesuai dengan waktu yang telah ditentukan. Inovasi ini bertujuan untuk menghemat pemakaian listrik dan mencegah terjadinya kerusakan pada perangkat elektronika yang diakibatkan oleh terlalu lama mencolok pada stop kontak, serta membantu memaksimalkan penggunaan alat elektronik secara terjadwal dan teratur.

## 1.2 Rumusan Masalah

Dari latar belakang diatas, masalah yang dapat diidentifikasi adalah :

1. Bagaimana merancang dan membangun *timer* ON/OFF agar stop kontak dapat hidup atau mati sesuai keperluan / kebutuhan?
2. Bagaimana merancang dan membangun *timer* ON/OFF yang dapat berjalan secara berkala / teratur sesuai jadwal yang telah ditentukan?

## 1.3 Tujuan

Tujuan dari penelitian yang akan dilakukan adalah sebagai berikut :

1. Merancang dan membangun *timer* ON/OFF agar stop kontak dapat hidup atau mati sesuai keperluan / kebutuhan
2. Merancang dan membangun *timer* ON/OFF yang dapat berjalan secara berkala / teratur sesuai jadwal yang telah ditentukan

#### 1.4 Batasan Masalah

1. Jumlah stop kontak yang di gunakan sebanyak tiga buah
2. Tidak membahas besar beban yang di kendalikan

#### 1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini secara Garis besar tersusun dari 5 (lima) bab, yaitu diuraikan sebagai berikut:

##### 1. BAB I PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

##### 2. BAB II LANDASAN TEORI

Pada Bab ini akan dibahas teori penunjang dari permasalahan, yaitu mengenai Arduino Mega 2560, LCD OLED, *Switch*, RTC (*Real Time Clock*) DS3231, *Relay*, stop kontak.

##### 3. BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM

Pada Bab ini akan dibahas tentang blog diagram sistem serta metode yang dilakukan dalam memutuskan arus listrik secara terkendali. Meliputi pembuatan algoritma untuk menjalankan alat, merancang tampilan menu pada layar LCD OLED, *flow cart software IDE* untuk pengaturan Arduino,

##### 4. BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil yang diperoleh dari proses penggabungan beberapa komponen sehingga dapat menghasilkan

output yang bisa di gunakan untuk memutus arus listrik. Bagaimana alat dapat mengalirkan listrik sesuai dengan yang di inginkan oleh pengguna.

## 5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan yang didapat dari hasil penelitian berdasarkan rumusan masalah serta saran untuk perkembangan penelitian selanjutnya.



## BAB II

### LANDASAN TEORI

#### 2.1 Arduino Mega 2560

Arduino mega 2560 adalah papan mikrokontroler berdasarkan ATmega2560 (*datasheet*). Ini memiliki 54 digital *Pin* input / output (*Pin* 15 dapat digunakan sebagai output PWM (*Pulse Width Modulation*)) , 16 analog input , 4 UART (*Universal Asynchronous Receiver Transmitter*) (hardware port serial) , osilator kristal 16 MHz , koneksi USB , jack listrik , header ICSP (*In-Circuit Serial Programming*) , dan tombol reset. Semuanya diperlukan untuk mendukung kerja mikrokontroler, cara mengaktifkan Arduino mega 2560 adalah dengan menghubungkannya ke komputer dengan kabel USB atau memberikan *power* dengan adaptor AC - DC atau baterai. Arduino Mega ini *compatible* dengan Arduino Duemilanove atau Diecimila.

Mega 2560 adalah update dari Arduino Mega. Mega 2560 berbeda dari semua *board* sebelumnya yang tidak menggunakan FTDI (*Future Technology Devices International*) chip driver USB - to -serial. Revisi ke 2 dari *board* Mega 2560 memiliki resistor 8U2, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU (*Device Firmware Update*).

Revisi 3 dari *board* Mega 2560 memiliki fitur-fitur baru berikut :

- 1.0 *Pinout* : menambahkan SDA (*Shouldbe Dhe Ather*) dan *Pin* SCL (*Simply Connect Last*) yang dekat dengan *Pin* AREF dan dua *Pin* baru lainnya ditempatkan dekat dengan *Pin* RESET , IOREF yang memungkinkan *shield* untuk beradaptasi dengan tegangan yang tersedia dari *board*. Nantinya , *shield* akan *compatible* baik dengan *board* yang

menggunakan AVR (*Advanced Versatile RISC*) , yang beroperasi dengan 5V dan dengan Arduino Due yang beroperasi dengan 3.3V. Yang kedua adalah *Pin* tidak terhubung , yang disediakan untuk tujuan lainnya.

- Sirkuit RESET kuat .
- Atmega 16U2 menggantikan 8U. (Arduino.cc)



**Gambar 2.1** Arduino Mega 2560 Sisi Depan (Kiri) dan Belakang (Kanan)

Secara umum arduino terdiri dari dua bagian, yaitu:

1. *Hardware*: papan input/output (I/O)
2. *Software*: *software* arduino meliputi IDE untuk menulis program, driver untuk koneksi dengan komputer, contoh program dan *library* untuk pengembangan program. (Djuandi, 2011)

Berikut adalah Tabel 2.1 spesifikasi dari arduino mega 2560.

**Tabel 2.1** Spesifikasi Arduino Mega 2560

Mikrokontroler	ATMega 2560
Tegangan Operasi	5V
<i>Input</i> tegangan(rekomendasi)	7 – 12V
<i>Input</i> tegangan (Maksimal)	6 – 20V
<i>Digital I/O Pin</i>	54 (15 <i>Pin</i> PWM)
<i>Pin input Analog</i>	16
DC current per I/O <i>Pin</i>	40mA
<i>Pin</i> DC Current untuk 3.3V	50Ma
Memori <i>flash</i>	256Kb, 8Kb digunakan untuk <i>bootloader</i>

SRAM	8Kb
EEPROM	4Kb
<i>Clock speed</i>	16 Mhz

## 1. Daya (*Power*)

Arduino mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Eksternal ( non - USB ) daya dapat berasal baik dari adaptor AC - DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug 2.1mm pusat - positif ke colokan listrik *board*. Baterai dapat dimasukkan dalam Gnd dan Vin *Pin* header dari konektor daya.

*Board* dapat beroperasi pada pasokan eksternal 6 sampai 20 volt. Jika tegangan dengan kurang dari 7V , tegangan pada *board* kemungkinan akan tidak stabil. Jika menggunakan lebih dari 12V , regulator tegangan bisa panas dan merusak *board*. Kisaran yang disarankan adalah 7 sampai 12 volt .

*Pin* listrik adalah sebagai berikut :

VIN : Tegangan *input* ke *board* Arduino ketika itu menggunakan sumber daya eksternal ( ebagai lawan 5 volt dari koneksi USB atau sumber daya diatur lain). Kita dapat memasok tegangan melalui *Pin* ini.

5V : *Pin output* 5V diatur dari regulator di *board*. *Board* dapat diaktifkan dengan daya baik dari colokan listrik DC (7 - 12V) , konektor USB (5V) , atau *Pin* VIN dari board (7-12V). Jika tegangan diberika melalui 5V atau 3.3V melewati regulator , dan dapat merusak *board*, maka tidak disarankan.

3V3 : Sebuah pasokan 3,3 volt dihasilkan oleh regulator on-board yang dapat menarik arus maksimum 50 mA.

GND : *Pin* tanah.

IOREF : *Pin* pada *board* Arduino memberikan tegangan referensi saat mikrokontroler sedang beroperasi. Sebuah *shield* dikonfigurasi dengan benar agar dapat membaca *Pin* tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada *output* untuk bekerja dengan 5V atau 3.3V.

(arduino.cc)

## 2. Memori

ATmega2560 memiliki 256 KB dari *flash memory* untuk menyimpan kode (8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM ( yang dapat dibaca dan ditulis dengan *library* EEPROM ). (Arduino.cc)

## 3. Input dan Output

Masing-masing dari 54 digital *Pin* (*Pin header*) pada Mega dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi dari *PinMode*( ), *digitalWrite*( ), dan *digitalRead*( ). Mereka beroperasi pada tegangan 5V. Setiap *Pin* dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up internal* yang (terputus secara *default*) dari 20-50 KOhms. Selain itu, beberapa *Pin* memiliki fungsi khusus :

**Serial :** 0(RX) dan 1(TX); Serial 1 : 19(RX) dan 18(TX) ; Serial 2 : 17(RX) dan 16 (TX) ; Serial 3 : 15 (RX) dan 14(TX). Yang digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. *Pin* 0 dan 1 juga terhubung ke *Pin* yang sesuai dari ATmega16U2 USB - to- TTL *chip* Serial. Letak *PIN* serial dapat dilihat pada gambar 2.25.

**Eksternal Interupsi:** 2 (*interrupt* 0) , 3 (*interrupt* 1) , 18 (*interrupt* 5) , 19 (*interrupt* 4) , 20 (*interrupt* 3) , dan 21 (*interrupt* 2). *Pin* ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, naik atau jatuh tepi, atau perubahan nilai. Lihat *AttachInterrupt()* fungsi untuk rincian.

**PWM:** *Pin* 2-13 dan 44 sampai 46. Menyediakan 8 - bit PWM *output* dengan *analogWrite()* *function*. Letak *PIN* PWM dapat dilihat pada gambar 2.25.

**SPI (*Serial Peripheral Intervace*) :** 50(MISO), 51(MOSI), 52(SCK), 53(SS). *Pin* ini mendukung komunikasi SPI menggunakan *library* SPI. *Pin* SPI juga pecah pada header ICSP, yang secara fisik kompatibel dengan Uno , Duemilanove dan Diecimila. Letak *PIN* SPI dapat dilihat pada gambar 2.25.

**LED :** 13. Ada *built -in* LED terhubung ke *Pin* digital 13. Ketika *Pin* dengan nilai TINGGI , LED menyala , ketika *Pin* yang dipakai RENDAH , lampu akan mati.

**TWI (*Two – Ware Inteerface*) :** *Pin* 20(SDA) dan *Pin* 21(SCL). Dukungan komunikasi TWI menggunakan *library* Wire. Yang perlu diperhatikan

adalah bahwa *Pin* ini tidak berada di lokasi yang sama dengan *Pin* TWI pada Duemilanove atau Diecimila.

Arduino Mega2560 memiliki 16 input analog, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default* mereka mengukur dari *ground* sampai 5 volt, meskipun mungkin untuk mengubah jangkauan menggunakan *Pin* AREF dan fungsi `analogReference()`.

## 2.2 LCD OLED

LCD OLED merupakan salah satu jenis media *output* yang dapat digunakan oleh module Arduino atau *controller* lain untuk menghasilkan sebuah tampilan. LCD OLED digunakan karena memiliki kelebihan yaitu kontras pixelnya yang sangat tajam dan tidak memerlukan cahaya *backlight* sehingga dalam hal konsumsi daya lebih hemat. Namun kekurangan dari LCD OLED ini adalah ukurannya yang terbilang lebih kecil daripada LCD TFT/ LCD *Graphic*.



**Gambar 2.2** LCD OLED

## 2.3 Switch

*Push button Switch* (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Contoh gambar dapat

di lihat pada Gambar 2.3. Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai *device* penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.



**Gambar 2.3** *Switch*

*Push button Switch* hanya memiliki kondisi *On* dan *Off* (1 dan 0). Istilah *On* dan *Off* dianggap penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi *On* dan *Off*. *Push button Switch* menjadi *device* yang paling umum digunakan untuk memulai dan mengakhiri kerja pada mesin industri karena sistem kerjanya yang *unlock* dan langsung berhubungan dengan operator. Sehingga bisa dipastikan secanggih apapun peralatan mesin, pasti membutuhkan saklar seperti *push button Switch* atau perangkat lain yang sejenis yang digunakan untuk mengatur pengkondisian *On* dan *Off* aliran listrik.

Berdasarkan fungsi kerjanya, *push button Switch* mempunyai 2 tipe kontak yaitu *NC (Normally Close)* dan *NO (Normally Open)*.

1. *NO (Normally Open)*, merupakan kontak terminal dengan kondisi normal terbuka (aliran arus listrik tidak mengalir), dan menutup (*Close*) ketika tombol saklar ditekan. Sehingga saklar akan mengalirkan atau menghubungkan arus

listrik ketika tombol ditekan. Kontak *NO* biasa digunakan sebagai penghubung atau untuk menyalakan sistem *circuit* (*Push Button ON*).

2. *NC* (*Normally Close*), merupakan kontak terminal dengan kondisi normal tertutup (mengalirkan arus listrik), dan membuka (*Open*) ketika tombol saklar ditekan. Sehingga saklar akan memutuskan aliran arus listrik ketika saklar ditekan. Kontak *NC* biasa digunakan sebagai pemutus atau mematikan sistem *circuit* (*Push Button Off*).

#### 2.4 RTC (*Real Time Clock*) DS3231

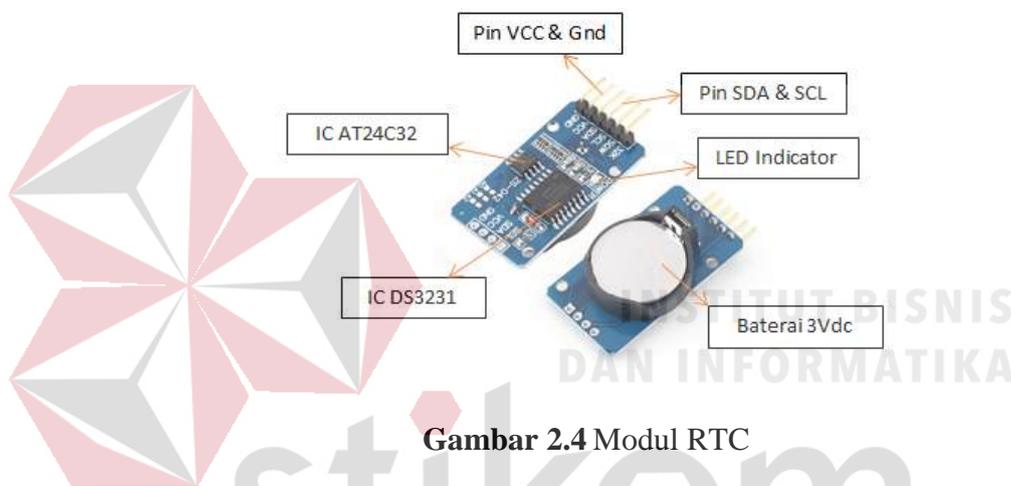
*Module* RTC DS3231 merupakan salah satu jenis *module* yang berfungsi sebagai penghitung dan menjaga/menyimpan data waktu secara *real time*. Selain itu terdapat pula IC EEPROM tipe AT24C32 yang dapat dimanfaatkan juga pada *module* ini.

*Interface* yang digunakan untuk mengakses modul ini menggunakan I2C atau *two wire* (SDA dan SCL). Sehingga jika diakses menggunakan mikrontroler, sebagai contoh Arduino Uno, *Pin* yang dibutuhkan 2 *Pin* saja dan tambahan 2 *Pin* untuk *power*. *Module* DS3231 RTC ini pada umumnya sudah satu paket dengan *battery* CR2032 3V yang berfungsi sebagai *back up* RTC apabila catudaya utama mati.

Dibandingkan dengan RTC DS1302, RTC DS3231 ini memiliki banyak kelebihan. Salah satunya untuk *range VCC input* dapat *disupply* menggunakan tegangan antara 2.3V sampai 5.5V dan memiliki cadangan baterai.

Berbeda dengan DS1307, pada DS3231 memiliki kristal yang terintegrasi (sehingga tidak diperlukan kristal eksternal), sensor suhu, 2 alarm waktu terprogram, *Pin* output 32.768 kHz untuk memastikan akurasi yang lebih tinggi.

Selain itu, terdapat juga EEPROM AT24C32 yang memiliki 32K EEPROM untuk menyimpan data, sehingga sesuai digunakan untuk aplikasi yang memerlukan fitur *data logging* dengan presisi waktu yang lebih tinggi.



**Gambar 2.4** Modul RTC

Spesifikasi dan fitur :

- Tegangan *Supply* : DC 2.3V – 5.5V
- Komunikasi : I2C atau SDA, SCL
- Kecepatan pengiriman data (*I2C Interface*) : 400kHz
- *Memory chips*: AT24C32 dengan kapasitas penyimpanan 32K
- RTC yang sangat akurat untuk mengelola semua fungsi pengatur waktu
- Ketepatan Jam: Kisaran 0-40, akurasi 2ppm, kesalahan sekitar 1 menit
- Jam *real time* menghitung detik, menit, jam, tanggal, bulan, bulan, hari dalam seminggu, dan tahun, dengan *range* tahun berlaku hingga 2100.

## 2.5 Relay

*Relay* adalah salah satu komponen elektronika yang berupa saklar atau *Switch* elektrik dan dioperasikan menggunakan tenaga listrik sebagai sumber energi. *Relay* juga biasa disebut sebagai komponen elektromekanikal yang terdiri dari dua bagian utama yaitu *coil* atau elektromagnet dan kontak saklar atau mekanikal.

Komponen *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan sejumlah kontaktor yang tersusun atau sebuah kontak saklar yang dapat dikendalikan oleh rangkaian elektronik lainnya, sehingga dengan menggunakan arus listrik yang kecil, dapat mengalirkan arus listrik bertegangan lebih tinggi. Gambar 2.5 adalah gambar dan juga simbol dari komponen *Relay*.



**Gambar 2.5** Gambar *Relay* dan Simbol *Relay*

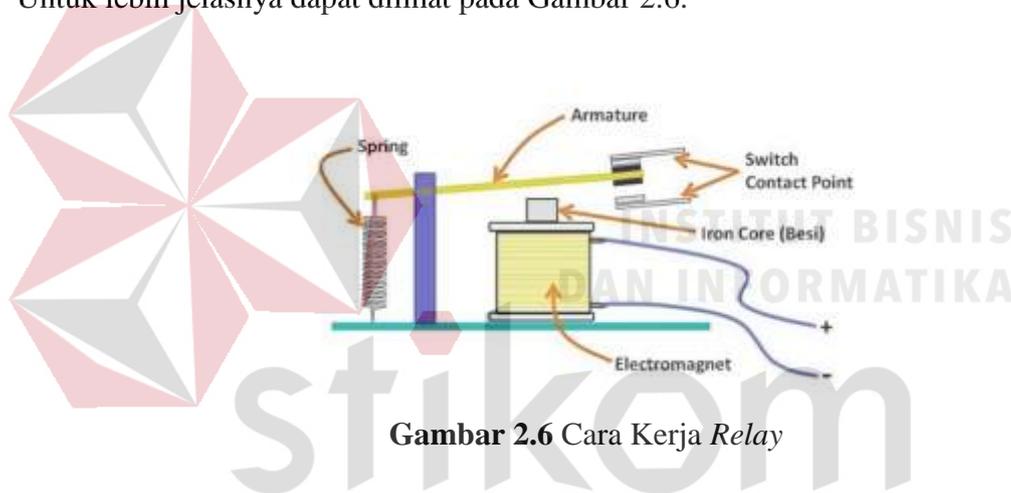
### Fungsi *Relay*

Seperti yang telah dijelaskan bahwa *Relay* memiliki fungsi sebagai saklar elektrik. Namun apabila diaplikasikan ke dalam rangkaian elektronika, *Relay* memiliki beberapa fungsi yang cukup unik. Berikut adalah beberapa fungsi komponen *Relay* saat diaplikasikan ke dalam sebuah rangkaian elektronika.

1. Mengendalikan sirkuit tegangan tinggi dengan menggunakan bantuan *signal* tegangan rendah.
2. Menjalankan fungsi logika atau *logic function*
3. Memberikan fungsi penundaan waktu atau *time delay function*

### Cara Kerja Relay

Perlu di ketahui bahwa *Relay* memiliki 4 buah bagian penting yaitu *Electromagnet (Coil)*, *Armature*, *Switch Contact Point* (Saklar), dan *Spring*. Untuk lebih jelasnya dapat dilihat pada Gambar 2.6.



Gambar 2.6 Cara Kerja Relay

Dari Gambar 2.6. sebuah besi (*Iron Core*) yang dililit oleh kumparan (*Coil*) berfungsi untuk mengendalikan besi tersebut. Apabila kumparan *Coil* teraliri arus listrik, maka akan muncul gaya elektromagnetik yang dapat menarik *Armature* sehingga dapat berpindah dari posisi yang sebelumnya tertutup (NC) menjadi terbuka (NO).

Dalam posisi NO, saklar dapat mengalirkan arus listrik. Namun pada saat tidak teraliri arus listrik, *Armature* akan kembali ke posisi awal (NC). Sedangkan *Contact Point* (penghubung) adalah sebagai saklar yang pergerakannya bergantung pada ada atau tidaknya arus listrik pada *Coil*.

## 2.6 Stop Kontak

Stop kontak merupakan salah satu komponen listrik yang berfungsi sebagai muara penghubung antara arus listrik dengan peralatan listrik. Agar alat listrik dapat terhubung dengan stop kontak, maka diperlukan kabel dan steker untuk ditancapkan pada stop kontak. Gambar 2.7 merupakan contoh gambar stop kontak.

Berdasarkan bentuk serta fungsinya, stop kontak dapat dibedakan menjadi dua jenis, yaitu:

1. Stop kontak kecil, merupakan stop kontak yang memiliki dua lubang (kanal) yang berfungsi sebagai penyalur listrik pada daya rendah ke alat - alat listrik melalui steker yang berjenis kecil juga.
2. Stop kontak besar, merupakan stop kontak dengan dua kanal AC yang dilengkapi dengan lempeng logam pada sisi atas dan bawah kanal AC yang berfungsi sebagai *ground*. Saklar jenis ini biasa digunakan untuk daya yang lebih besar.

Sedangkan berdasarkan tempat pemasangannya, stop kontak dibedakan dalam dua jenis, yaitu:

1. Stop kontak *in bow*, merupakan stop kontak yang dipasang di dalam tembok.
2. Stop kontak *out bow*, merupakan stop kontak yang dipasang di luar tembok atau hanya diletakkan di permukaan tembok pada saat berfungsi sebagai stop kontak *portable*.



**Gambar 2.7** Gambar Stop Kontak



## BAB III

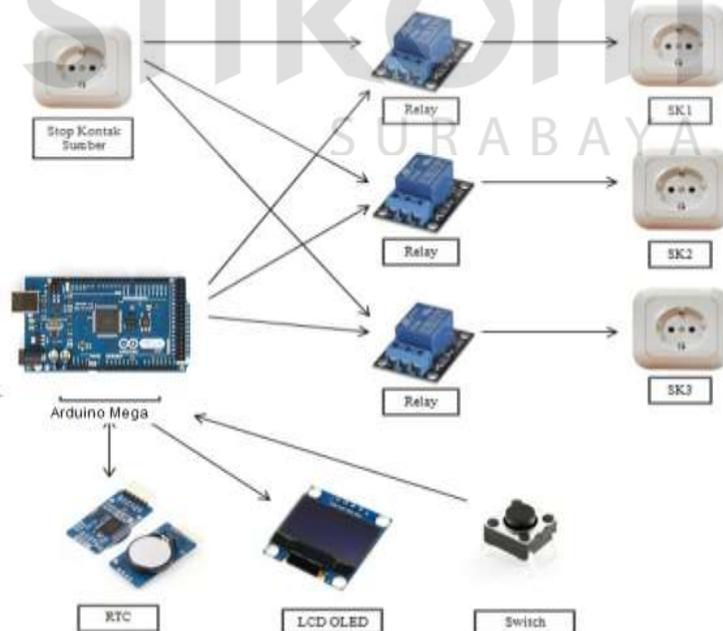
### METODE PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan meliputi studi literatur dan penelitian alat. Studi literatur dilakukan untuk mencari teori atau informasi dari buku, jurnal, dan artikel-artikel yang berkaitan dengan permasalahan, terutama metode penelitian yang akan digunakan. Dari informasi studi literatur yang diperoleh, maka dilakukan penelitian alat, yaitu perancangan alat yang akan di kerjakan dalam tugas akhir ini

#### 3.2 Model Perancangan

Pada perancangan ini penulis menggambarkan perancangan sistemnya seperti pada Gambar 3.1 berikut.



**Gambar 3.1** Gambar Perancangan Alat

Dari Gambar 3.1 di dapatkan bahwa setiap komponen memiliki tugas dan fungsi yang berbeda – beda seperti berikut :

Stop kontak disini ada empat buah. Pertama adalah stop kontak sumber, dimana stop kontak ini merupakan pusat dimana aliran listrik terhubung ke semua perangkat. Stop kontak kedua sampai keempat di inisialisasi menjadi “SK1”, “SK2” dan “SK3”. Ketiga stop kontak ini adalah stop kontak terkendali yang *On/Off* nya diatur oleh *mikrokontroler*.

*Relay* disini berfungsi sebagai penghubung antara stop kontak sumber dan stop kontak terkendali. Serta juga berfungsi sebagai pemutus aliran listrik antara kedua buah stop kontak. *Relay* ini dapat bekerja atas perintah dari Arduino yang telah di sesuaikan.

Arduino ini sendiri merupakan otak dari semua satu kesatuan alat ini. Dimana di dalam Arduino ini di berikan program yang dapat mengatur kapan *Relay* akan bekerja, mematikan *Relay* sesuai yang di inginkan, mengatur waktu dengan menggunakan RTC, bisa menampilkan pilihan menu dengan bantuan LCD.

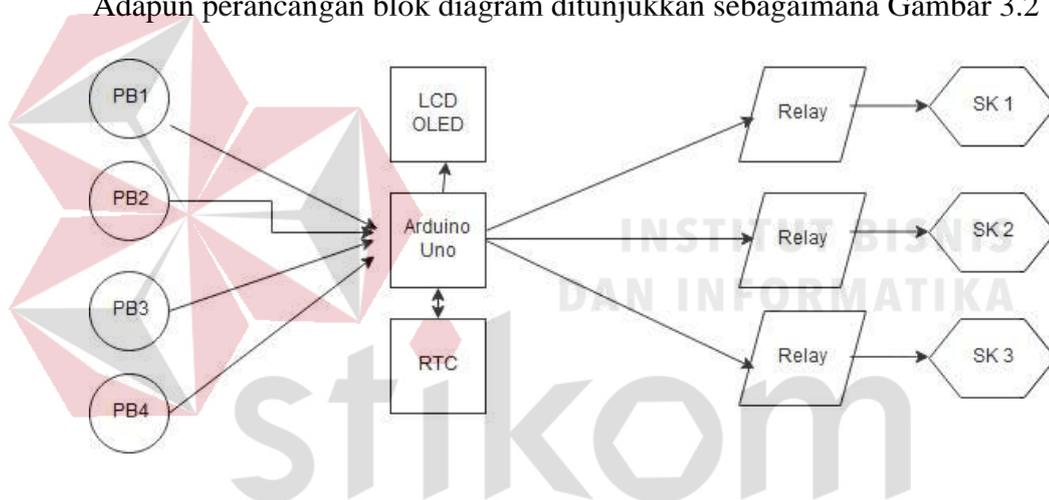
RTC memberikan informasi mengenai waktu. Di dalam projek ini RTC berfungsi sebagai pengatur waktu yang akan di inputkan sesuai kebutuhan pengguna dimana pengaturannya sendiri dilakukan di dalam Arduino.

Untuk menampilkan menu yang tersedia di dalam alat ini, maka yang memiliki peranan utama adalah LCD OLED. Dengan menggunakan komponen ini pengguna dapat mengetahui ada pilihan menu apa saja yang dapat di akses serta pengguna dapat mengetahui alat ini dapat bekerja atau tidak dengan mengetahui dari layar LCD OLED yang menyala dan menampilkan menu yang tersedia.

*Push Button* disini bukan berfungsi sebagai pemutus aliran listrik, melainkan sebagai alat bantu yang akan digunakan oleh pengguna untuk memilih menu yang tampil pada layar LCD. Misal ada dua buah menu pilihan yang tampil pada layar, maka *push button* akan membantu untuk memilih menu yang akan di pilih oleh pengguna. Selain itu *push button* juga membantu pengguna untuk memasukkan waktu yang akan di gunakan atau di inginkan.

### 3.3 Perancangan Sistem

Adapun perancangan blok diagram ditunjukkan sebagaimana Gambar 3.2 :

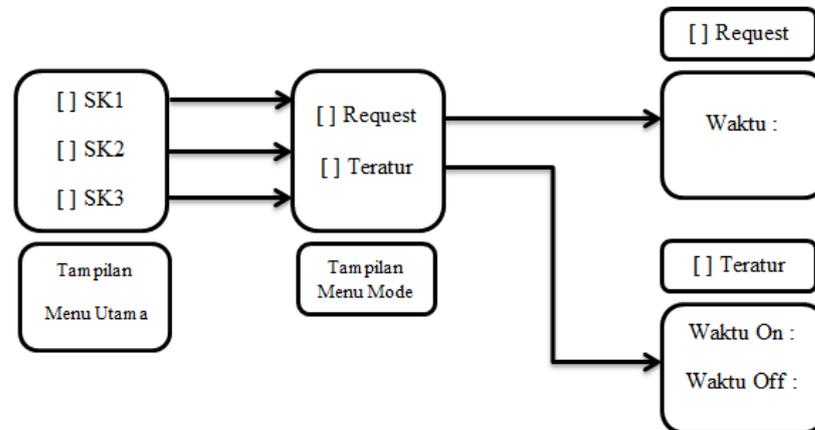


Gambar 3.2 Blok Diagram Sistem

Dalam tugas akhir ini penulis hanya akan fokus pada bagaimana aliran listrik dapat tersalurkan dan terputus kepada stop kontak yang di inisialisasi dengan “SK1, SK2 dan SK3”. Inisialisasi “PB” memiliki arti *push button* dimana komponen ini berfungsi sebagai alat bantu pemberi input kepada alat yang kemudian di olah oleh Arduino yang dimana akan di tampilkan pada layar LCD OLED yang akan menampilkan beberapa menu pilihan yang dapat di pilih oleh pengguna.

### 3.4 Rencana Menu Yang Akan di Tampilkan Pada Layar LCD OLED

Adapun perancangan menu akan ditunjukkan pada Gambar 3.3:



**Gambar 3.3** Gambar Rancangan Menu

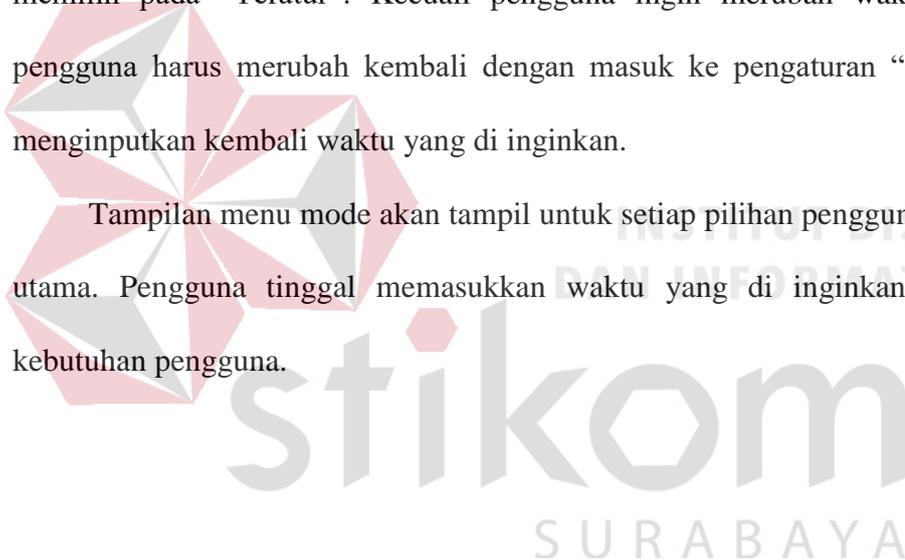
Di dalam rancangan ini nantinya LCD akan mempunyai atau akan menampilkan pilihan menu utama pada saat alat dinyalakan yang bisa dipilih oleh pengguna sebanyak tiga buah pilihan menu. Yaitu menu “SK1”, “SK2” dan “SK3”. Menu “SK1” sampai “SK3” disini ialah pengganti dari kata stop kontak 1 sampai stop kontak 3. Pada saat menu utama tersebut tampil pada layar LCD, pengguna di harapkan memilih salah satu dari menu untuk di beri pengaturan secara langsung. Setelah pengguna memilih salah satu pilihan pada menu utama, selanjutnya pengguna akan masuk ke tampilan menu mode, dimana pengguna diminta untuk memasukkan pilihan mode yang diinginkan.

Pilihan mode pertama yaitu “Request” yang dimana di dalam menu ini pengguna di minta untuk memasukkan lama waktu yang di inginkan untuk mengalirkan listrik pada stop kontak sesuai kebutuhan. Misalkan saja pengguna ingin mengalirkan listrik pada stop kontak selama dua jam, maka pengguna

tinggal memasukkan angka 120 pada kolom waktu setelah memilih menu “Request” pada menu mode.

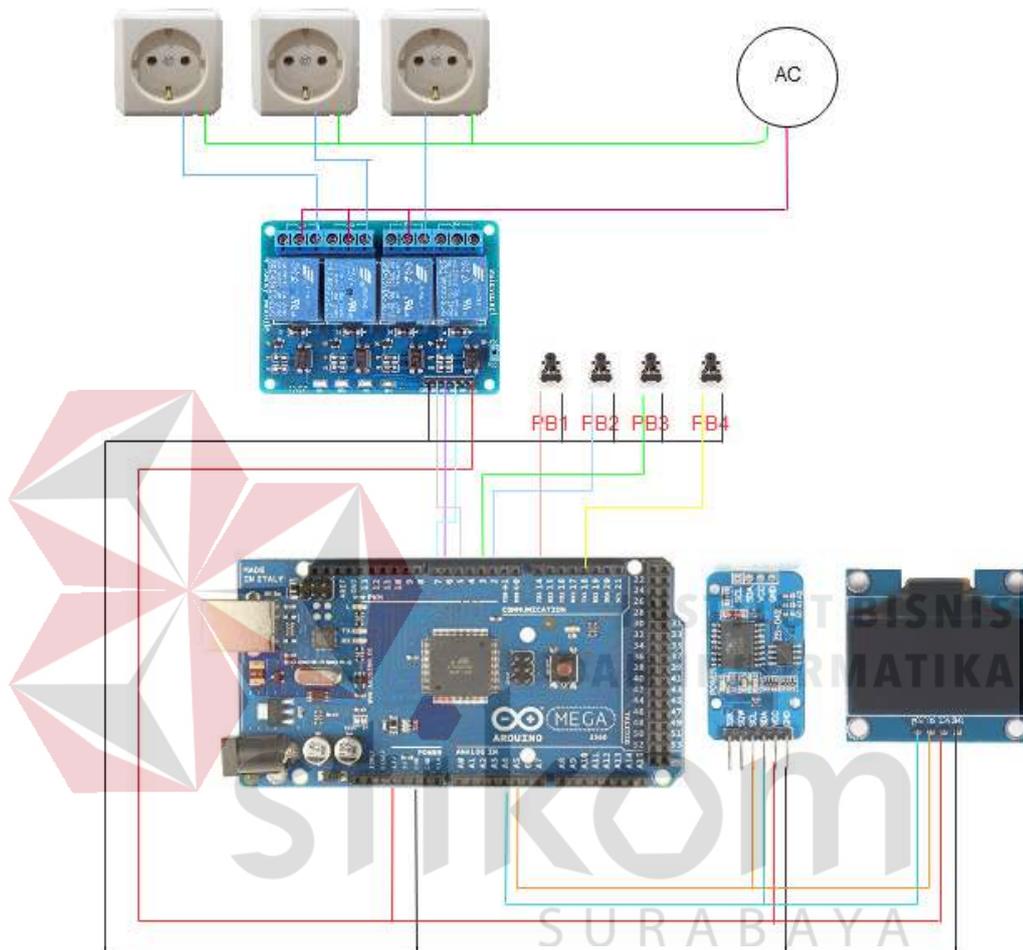
Pilihan mode kedua adalah “Teratur” dimana di dalam menu ini pengguna tetap memasukkan nilai waktu yang di inginkan, tetapi bedanya pada menu ini pengguna memasukkan nilai waktu yang di inginkan dengan cara pengguna memasukkan waktu mulai (waktu *on*) dan waktu akhir (waktu *off*). Dan bila pengguna ingin menggunakan menu ini pada waktu berikutnya, maka pengguna tidak perlu lagi memasukkan nilai waktu dan langsung menyalakan alat kemudian memilih pada “Teratur”. Kecuali pengguna ingin merubah waktunya, maka pengguna harus merubah kembali dengan masuk ke pengaturan “Teratur” dan menginputkan kembali waktu yang di inginkan.

Tampilan menu mode akan tampil untuk setiap pilihan pengguna pada menu utama. Pengguna tinggal memasukkan waktu yang di inginkan saja sesuai kebutuhan pengguna.



### 3.5 Perancangan Perangkat Keras

Perancangan perangkat keras dalam tugas akhir ini di jelaskan pada Gambar 3.4 di bawah ini :



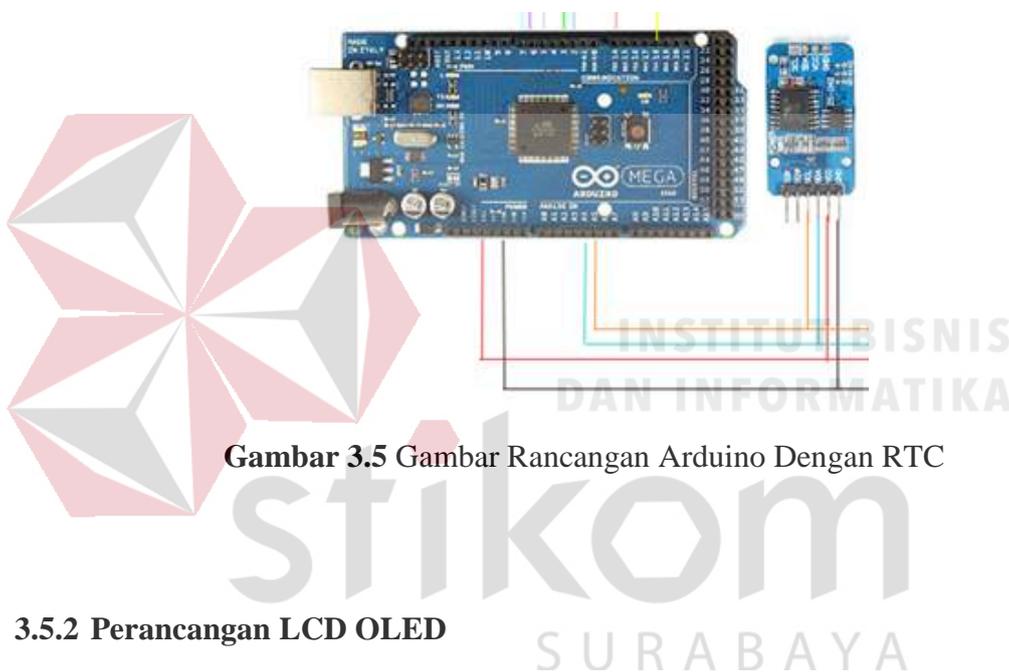
**Gambar 3.4** Gambar Rancangan Perangkat Keras

**Tabel 3.1** Tabel Hubungan *Pin* Modul Dengan *Pin* Arduino

No.	Modul	<i>Pin</i> Modul	<i>Pin</i> Arduino
1.	RTC	VCC	3,3 Volt
		GND	GND
		SCL	A5
		SDA	A4
2.	LCD OLED	VCC	3,3 Volt
		GND	GND
		SCL	A5
		SDA	A4
3.	Relay	VCC	3,3 Volt
		GND	GND
		IN3	5
		IN2	6
4.	PB1	IN1	7
		VCC	3,3 Volt
		GND	GND
5.	PB2	<i>Pin</i>	14
		VCC	3,3 Volt
		GND	GND
6.	PB3	<i>Pin</i>	2
		VCC	3,3 Volt
		GND	GND
7.	PB4	VCC	3,3 Volt
		GND	GND
		<i>Pin</i>	18
8.	AC	NC	NC
		GND	GND

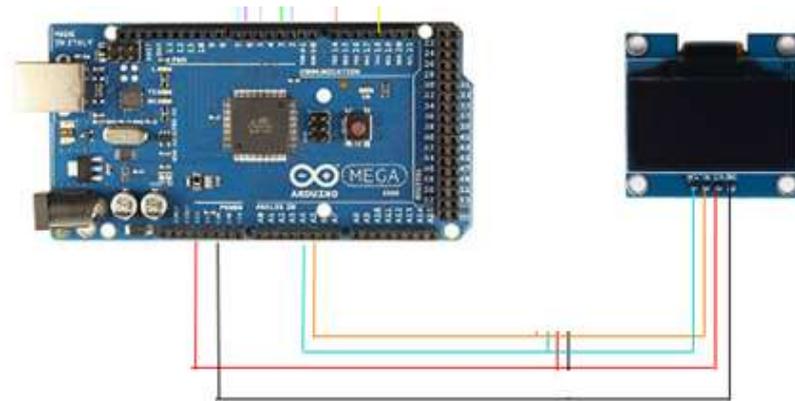
### 3.5.1 Perancangan RTC

Agar dapat memberi nilai waktu sesuai yang di inginkan oleh pengguna, maka dibutuhkan suatu komponen tambahan yang bernama RTC ( *Real Time Clock* ) yang terhubung langsung dengan Arduino uno. Dengan menggunakan komponen ini pengguna dapat memberi nilai masukan supaya alat ini dapat di gunakan sesuai kebutuhan pengguna. Adapun rancangan rangkaian RTC dapat dilihat pada Gambar 3.5

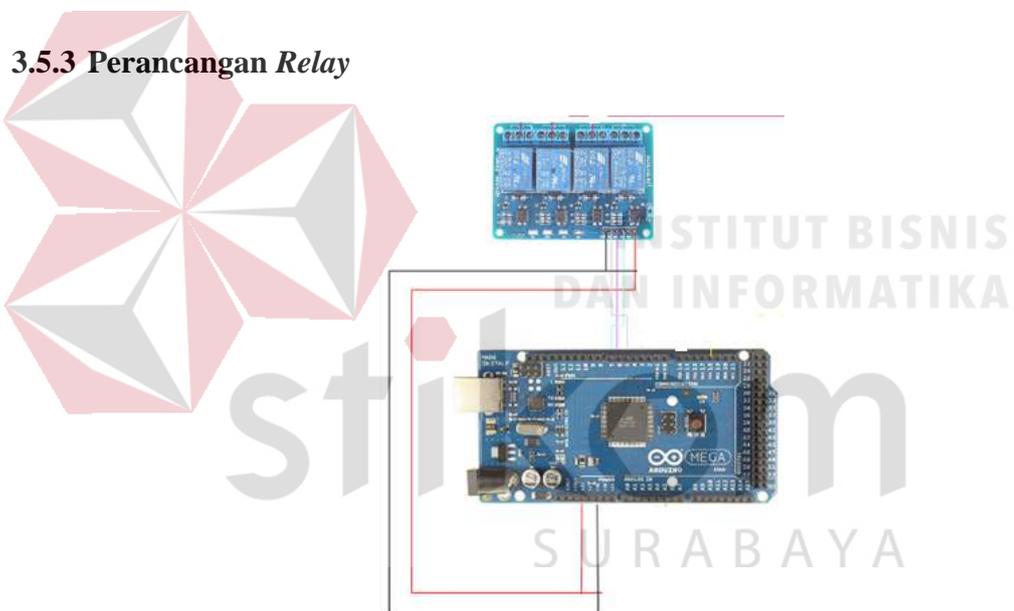


### 3.5.2 Perancangan LCD OLED

LCD OLED disini selain berfungsi sebagai indikator keberhasilan jalannya alat ini atau tidak, komponen ini juga berfungsi sebagai perantara agar pengguna dapat memberi nilai inputan dan dapat memilih menu yang tersedia didalam alat yang digunakan. Perananannya sangatlah penting, apabila komponen ini tidak menyala, maka pengguna tidak dapat melakukan apapun pada alat yang digunakan. Adapun rancangan rangkaian LCD OLED ditunjukkan pada Gambar 3.6



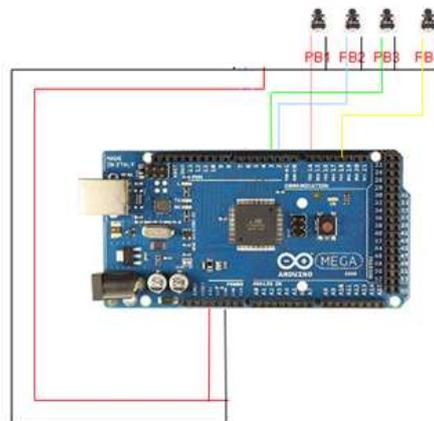
**Gambar 3.6** Gambar Rancangan Arduino Dengan LCD OLED



**Gambar 3.7** Gambar Rancangan Arduino Dengan *Relay*

*Relay* disini berfungsi sebagai pemutus aliran listrik antara alat dengan perangkat yang akan di hubungkan. Dengan menggunakan *Relay* maka pengguna dapat menggunakan alat seberapa lama akan digunakan untuk mengalirkan arus listrik. Dengan dihubungkan dengan Arduino, maka *Relay* dapat diatur sesuai kebutuhan. Adapun rancangan rangkaian *Relay* ditunjukkan pada Gambar 3.7.

### 3.5.4 Perancangan *Push Button*



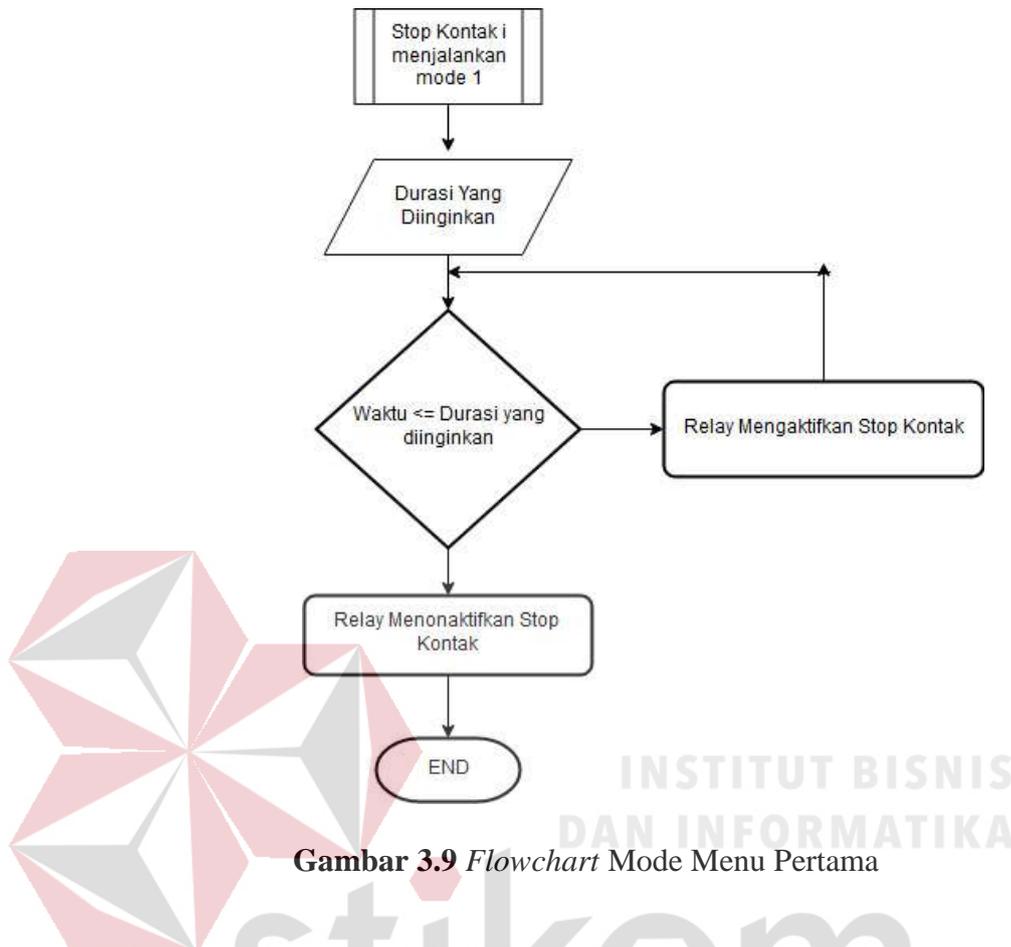
**Gambar 3.8** Gambar Rancangan Arduino Dengan Push Button

Dengan menggunakan *push button* maka pengguna dapat memberikan inputan pada alat yang digunakan. Selain itu komponen ini juga membantu pengguna untuk memilih menu yang di tampilkan pada layar LCD OLED kemudian melanjutkan langkah selanjutnya. Rancangan rangkaian *push button* ditunjukkan pada Gambar 3.8.

### 3.6 Perancangan Perangkat Lunak

Dari perancangan sistem diatas, selain perancangan *hardware*, juga dibutuhkan perancangan perangkat lunak untuk menjalankan perancangan *hardware* yang telah dibuat secara maksimal. Perangkat lunak terdiri dari beberapa algoritma perancangan dari sistem yang ditangani oleh pengontrol alias yang di atur oleh Arduino.

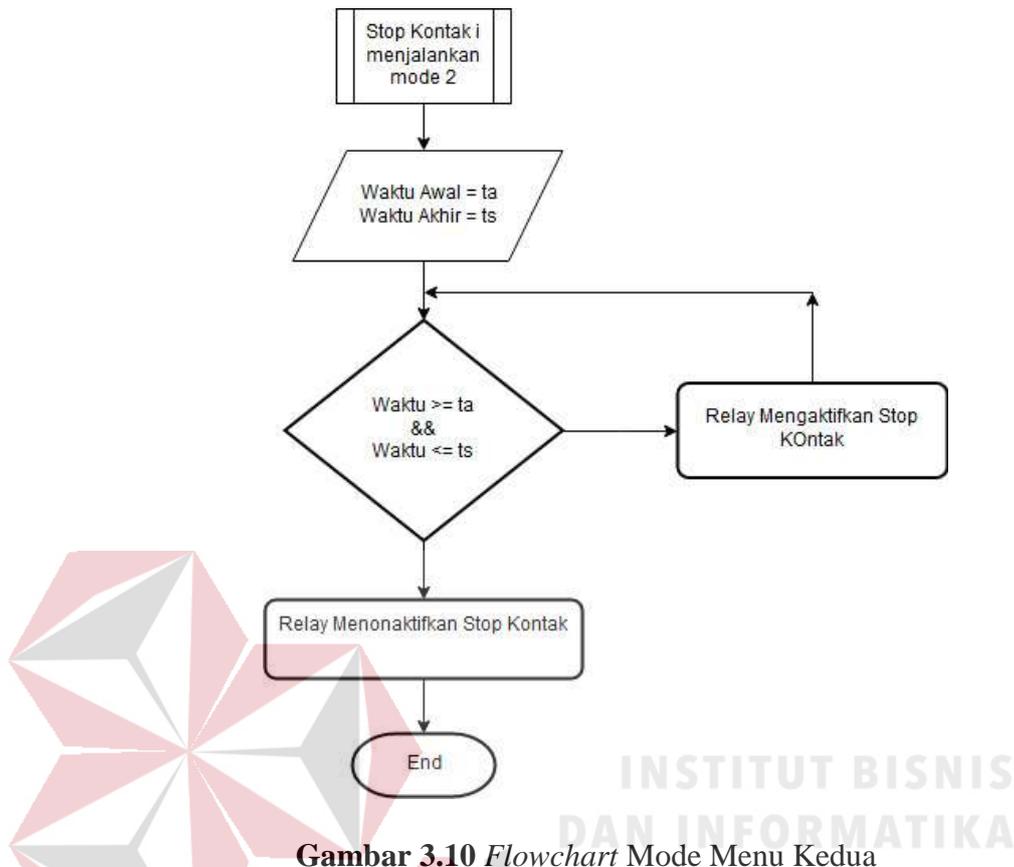
### 3.6.1 Flowchart Mode Pertama



**Gambar 3.9** Flowchart Mode Menu Pertama

Dalam mode pertama ini dijelaskan bahwa pengguna di arahkan untuk memberikan inputan berupa nilai waktu yang di inginkan oleh pengguna. Dimana pemberian waktu ini di harapkan akan berjalan sesuai dengan yang di inginkan oleh pengguna. Pertama–tama setelah masuk ke dalam mode ini, pengguna pastinya mengetahui yang akan di lakukan dalam mode ini yaitu harus menginputkan waktu seberapa lama alat tersebut akan di gunakan. Lalu jika sudah memberikan nilai, maka pengguna dapat mengaktifkan *Relay* yang dapat memutuskan arus listrik pada stop kontak terkendali yang sudah di atur. Namun apabila pengguna tidak memberikan nilai inputan, maka menu akan kembali kepada menu utama dan alat tidak dapat di gunakan sesuai kebutuhan dari pengguna. Alur dari mode ini dapat di lihat pada gambar 3.9.

### 3.6.2 Flowchart Mode Kedua



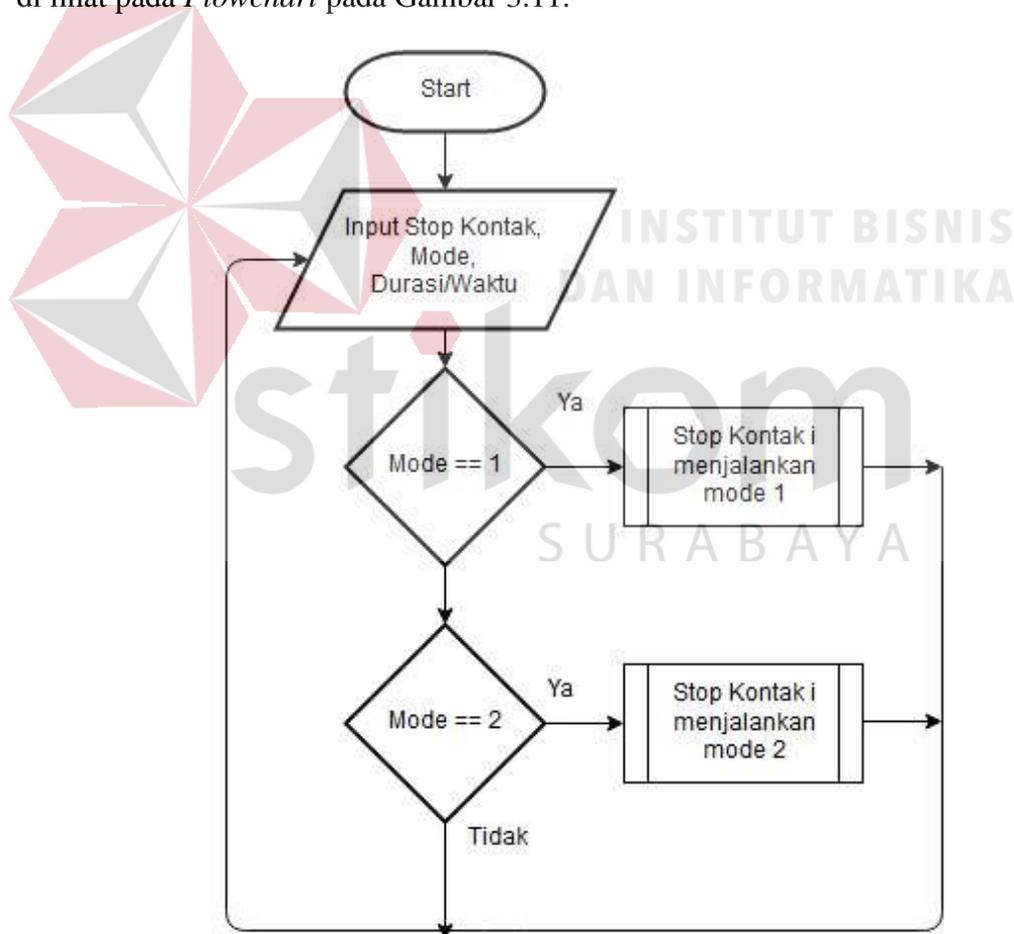
**Gambar 3.10** Flowchart Mode Menu Kedua

Mode kedua ini pengguna tetap menginputkan nilai waktu yang berfungsi untuk mengaktifkan berapa lama alat akan di gunakan. Namun bedanya dengan mode pertama ialah pengguna memasukkan waktu mulai (waktu *on*) dan waktu akhir (waktu *off*). Dan bila pengguna ingin menggunakan menu ini pada waktu berikutnya, maka pengguna tidak perlu lagi memasukkan nilai waktu dan langsung menyalakan alat saja. Kecuali pengguna ingin merubah waktunya, maka pengguna harus merubah kembali dengan masuk ke pengaturan menu ini dan menginputkan kembali waktu yang di inginkan.

Jadi di dalam menu ini inputan nilai yang sudah di masukkan ini dapat tersimpan sehingga pengguna tidak perlu memberi inputan nilai waktu kembali. Alur dari mode ini dapat di lihat pada *Flowchart* pada Gambar 3.10.

### 3.6.3 Flowchart Keseluruhan Sistem

Ini adalah gambaran keseluruhan dari alat ini. Dimana pertama kali saat alat di nyalakan maka pengguna akan di sajikan dua buah mode tampilan menu yang sudah di jelaskan pada penjelasan sebelumnya. Pengguna tinggal memilih saja akan menggunakan mode pertama atau mode kedua sesuai dengan keinginan dan kebutuhannya. Setelah memilih tentu saja pengguna tinggal melanjutkan saja alur dari setiap mode tersebut. Pengguna dapat menggunakan alat ini secara langsung setelah memilih kemudian memberi inputan nilai. Alur dari mode utama ini dapat di lihat pada *Flowchart* pada Gambar 3.11.



**Gambar 3.11** Flowchart Keseluruhan Sistem

### 3.7 Metode Analisis

Dalam tugas akhir ini analisa yang dilakukan adalah dengan melihat hasil dari beberapa komponen yang di rangkai :

1. Dengan melihat LCD OLED menyala atau tidak. Pada saat alat dinyalakan yang seharusnya terjadi adalah menyalnya LCD OLED, karena ini adalah indikator pertama alat ini dapat berjalan dengan baik atau sebaliknya. Apabila tidak menyala, maka dapat disebut terjadi kesalahan rangkaian atau terjadi konsleting pada LCD tersebut.
2. Menyalnya lampu indikator kecil pada setiap komponen. Misalnya adalah pada komponen Arduino, RTC, *Relay*. Jika pada salah satu komponen ada yang tidak menyala, maka komponen tersebut tidak dapat menjalankan fungsinya dengan baik sehingga keseluruhan alat ini tidak dapat berjalan dengan baik.
3. Melihat lampu LED yang menyala atau tidak. LED disini berfungsi sebagai indikator apabila *Relay* dapat bekerja menyalurkan listrik atau tidak. Apabila LED tidak menyala, maka *Relay* tersebut tidak bekerja mengalirkan listrik. Sebaliknya, apabila LED menyala tetapi pada stop kontak tidak dapat di aliri listrik, maka terjadi sebuah kesalahan sistem dari rangkaian tersebut.
4. Terakhir ialah melihat dari ke tiga stop kontak yang terhubung dengan *Relay*. Apabila *Relay* sudah menyala dan lampu LED sudah menyala juga, seharusnya stop kontak sudah teraliri listrik. Namun jika semua sudah menyala tetapi masih belum teraliri, maka perlu dilakukan analisa pada alat tersebut. Dimana mungkin terjadi suatu kesalahan pada saat merangkai alat atau terjadi kesalahan pada saat pemrograman.

## **BAB IV**

### **HASIL PENGUJIAN DAN PENGAMATAN**

Dalam bab ini penulis akan menguraikan dan menjelaskan beberapa hasil pengujian dari hasil penelitian tugas akhir ini. Pengujian yang dilakukan meliputi pengujian perangkat lunak (*software*) dan kinerja keseluruhan sistem.

#### **4.1. Pengujian Arduino**

Pengujian Arduino dilakukan dengan memasukan skrip *program* sederhana pada Arduino menggunakan aplikasi Arduino IDE. Arduino yang baik dapat menjalankan *program* dengan baik dan hasilnya akan terlihat langsung sesuai skrip *program* yang di masukkan.

##### **4.1.1 Tujuan**

Pengujian ini dilakukan untuk mengetahui apakah Arduino yang digunakan tidak mengalami kerusakan. Sehingga saat Arduino digunakan dapat membantu sistem berjalan dengan baik dan sesuai yang di inginkan.

##### **4.1.2 Alat Yang Di Gunakan**

Alat yang digunakan untuk melakukan pengujian antara lain :

- a. Kabel usb
- b. Arduino Mega 2560
- c. Komputer / laptop
- d. *Software* Arduino IDE

### 4.1.3 Prosedur Pengujian

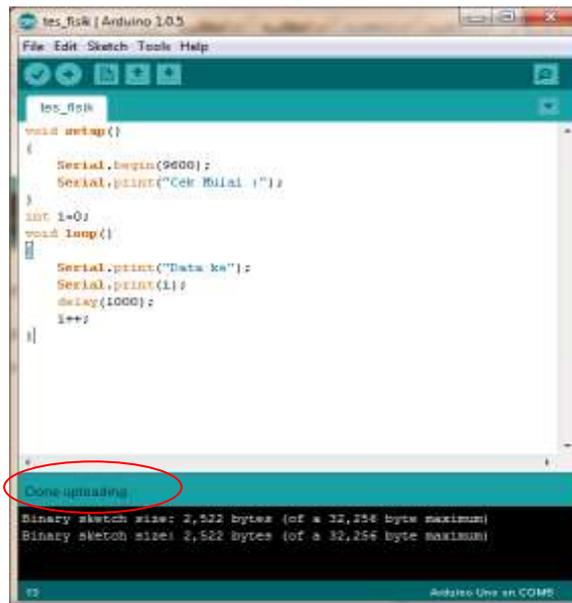
1. Hubungkan Arduino dengan kabel usb
2. Nyalakan komputer / laptop kemudian hubungkan kabel usb tadi dengan komputer.
3. Buka *software* Arduino IDE dan isi perintah dalam bahasa C. Sebagai contoh penulis memasukkan perintah sebagai berikut:

```
void setup()
{
    Serial.begin(9600);
    Serial.println("Cek Mulai :");
}
int i=0;
void loop()
{
    Serial.print("Data ke");
    Serial.println(i);
    delay(1000);
    i++;
}
```

4. Apabila telah selesai untuk mengisi perintah, maka tekan “*Verify*” untuk mengecek apabila terdapat perintah yang salah dalam bahasa C. Dan tekan “*Upload*” untuk memasukkan perintah tersebut ke dalam Arduino Mega.
5. Setelah *program* telah berhasil dimasukkan, maka tekan *icon Serial monitor* pada kanan atas. Maka akan muncul tampilan serial monitor.
6. Setelah window serial monitor muncul, amati kiriman data serial oleh Arduino.

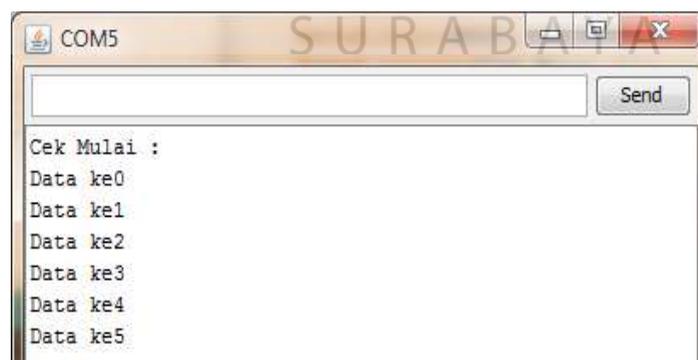
### 4.1.4 Hasil Pengujian

Hasil dari pengujian pengisian program ke Arduino dapat dilihat pada Gambar 4.1. Lingkaran merah menunjukkan bahwa Arduino yang digunakan berhasil di isi dengan *program* yang telah ditulis dalam *software* Arduino IDE.



**Gambar 4.1** *upload program berhasil*

Program yang dimasukkan kedalam Arduino merupakan *program* untuk mengirimkan data menggunakan serial. Proses pengiriman ini apabila Arduino masih dihubungkan dengan USB PC maka kita dapat menerima data yang dikirim menggunakan menu serial monitor pada *software* Arduino IDE. Hasil dari serial monitor dapat dilihat pada Gambar 4.1.



**Gambar 4.2** Program berhasil berjalan

Gambar 4.2 menunjukkan bahwa data dikirimkan sesuai dengan perintah *program* yang telah diisi pada Arduino. Dengan begitu Arduino ini dapat bekerja dengan baik, dan dapat digunakan untuk sistem.

## 4.2. Pengujian LCD OLED

Pengujian LCD Oled dilakukan dengan menggabungkan antara LCD Oled dengan Arduino uno dan kemudian memasukkan skrip *program* sederhana pada aplikasi Arduino IDE untuk mengetahui hasilnya.

### 4.2.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah LCD Oled yang digunakan tidak mengalami kerusakan. Sehingga saat digunakan dapat membantu menampilkan sesuatu yang di inginkan oleh pengguna.

### 4.2.2 Alat Yang Digunakan

Alat yang digunakan untuk melakukan pengujian antara lain :

- a. Kabel *usb*
- b. Arduino Uno
- c. LCD Oled
- d. Komputer / Laptop
- e. *Software* Arduino IDE
- f. Kabel *Jumper*

### 4.2.3 Prosedur Pengujian

- a. Hubungkan LCD Oled dengan Arduino uno menggunakan kabel *jumper*
- b. Nyalakan komputer / laptop kemudian hubungkan Arduino uno menggunakan kabel usb dengan komputer.

- c. Buka *software* Arduino IDE dan isi perintah dalam bahasa C. Sebagai contoh penulis memasukkan perintah sebagai berikut :

```
void setup() {
  u8g.setFont(u8g_font_unifont);
  u8g.setColorIndex(1);
}

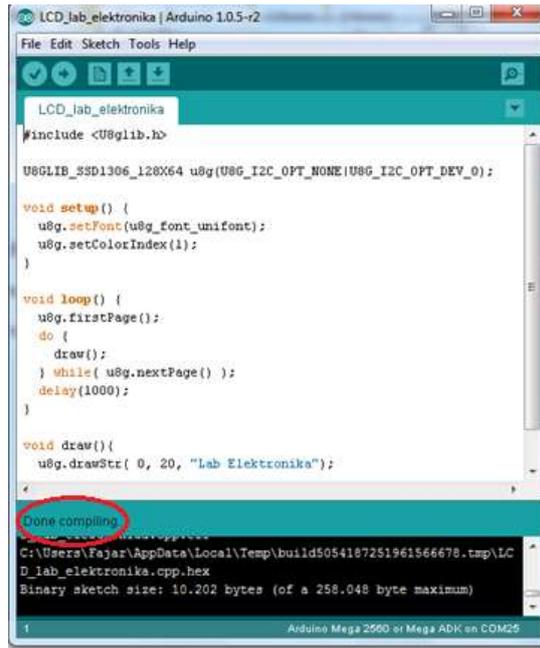
void loop() {
  u8g.firstPage();
  do {
    draw();
  } while( u8g.nextPage() );
  delay(1000);
}

void draw(){
  u8g.drawStr( 0, 20, "Lab Elektronika");
}
```

- d. Apabila telah selesai untuk mengisi perintah, maka tekan “*Verify*” untuk mengecek apabila terdapat perintah yang salah dalam bahasa C. Dan tekan “*Upload*” untuk memasukkan perintah tersebut ke dalam Arduino Uno.
- e. Setelah *program* telah berhasil dimasukkan, maka tekan *icon Serial monitor* pada kanan atas. Maka akan muncul tampilan serial monitor.
- f. Setelah window serial monitor muncul, amati kiriman data serial oleh Arduino.

#### 4.2.4 Hasil Pengujian

Hasil dari pengujian pengisian program ke Arduino dapat dilihat pada Gambar 4.3. Lingkaran merah menunjukkan bahwa Arduino yang digunakan berhasil di isi dengan *program* yang telah ditulis dalam *software* Arduino IDE.



```

LCD_lab_elektronika | Arduino 1.0.5-r2
File Edit Sketch Tools Help

LCD_lab_elektronika
#include <U8glib.h>

U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE|U8G_I2C_OPT_DEV_0);

void setup() {
  u8g.setFont(u8g_font_unifont);
  u8g.setColorIndex(1);
}

void loop() {
  u8g.firstPage();
  do {
    draw();
  } while( u8g.nextPage() );
  delay(1000);
}

void draw(){
  u8g.drawStr( 0, 20, "Lab Elektronika");
}

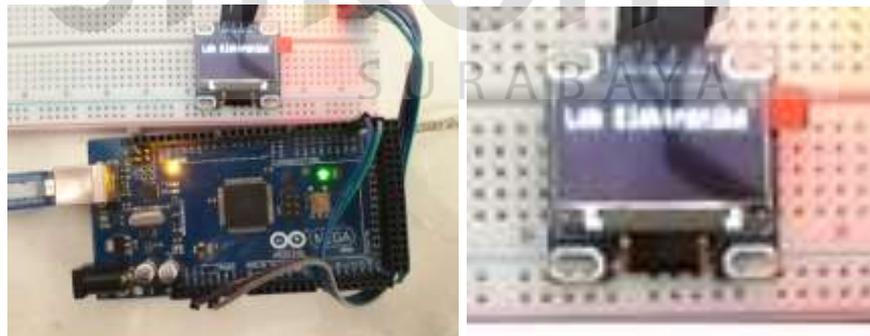
Done compiling
C:\Users\Fajar\AppData\Local\Temp\build5054187251961566678.tmp\LCD_lab_elektronika.cpp.hex
Binary sketch size: 10.202 bytes (of a 258.048 byte maximum)

Arduino Mega 2560 or Mega ADK on COM25

```

**Gambar 4.3** *upload program berhasil*

*Program* yang dimasukkan kedalam Arduino merupakan *program* untuk mencoba menampilkan suatu kalimat pada LCD Oled yang terdapat pada Gambar 4.3.



**Gambar 4.4** Program berhasil berjalan

Gambar 4.4 menunjukkan bahwa *program* dapat berjalan dan menampilkan sesuai yang di inginkan oleh pengguna pada LCD Oled.

### 4.3. Pengujian RTC (*Real Time Clock*)

Pengujian RTC dilakukan dengan menggabungkan antara RTC dengan Arduino uno dan kemudian memasukkan skrip *program* sederhana pada aplikasi Arduino IDE untuk mengetahui hasilnya.

#### 4.3.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah RTC yang digunakan tidak mengalami kerusakan. Sehingga saat digunakan dapat membantu menampilkan sesuatu yang di inginkan oleh pengguna.

#### 4.3.2 Alat yang digunakan

Alat yang digunakan untuk melakukan pengujian antara lain :

- a. Kabel USB
- b. Arduino Uno
- c. RTC ds3231
- d. Komputer / laptop
- e. *Software* Arduino IDE
- f. Kabel *Jumper*

#### 4.3.3 Prosedur Pengujian

- a. Hubungkan RTC dengan Arduino uno menggunakan kabel *jumper*
- b. Nyalakan komputer / laptop kemudian hubungkan Arduino uno menggunakan kabel usb dengan komputer.

c. Buka *software* Arduino IDE dan isi perintah dalam bahasa C. sebagai

contoh penulis memasukkan perintah sebagai berikut :

```

Time t;

int clockCenterX=31;
int clockCenterY=31;
int oldsec=0;

void setup()
{
  myOLED.begin();
  myOLED.setFont(Sinclair_M);

  rtc.begin();

  Serial.begin(115200);
  Serial.println("Send any character to enter serial
mode...");
  Serial.println();

  t = rtc.getTime();
}

void drawDisplay()
{
  // Clear screen
  myOLED.clrScr();

  // Draw Clockface
  for (int i=0; i<2; i++)
  {
    myOLED.drawCircle(clockCenterX, clockCenterY, 31-i);
  }
  for (int i=0; i<3; i++)
  {
    myOLED.drawCircle(clockCenterX, clockCenterY, i);
  }

  // Draw a small mark for every hour
  for (int i=0; i<12; i++)
  {
    drawMark(i);
  }
  t = rtc.getTime();
}

void drawMark(int h)
{
  float x1, y1, x2, y2;

  h=h*30;
  h=h+270;

  x1=29*cos(h*0.0175);
  y1=29*sin(h*0.0175);
  x2=26*cos(h*0.0175);

```

```

        y2=26*sin(h*0.0175);

myOLED.drawLine(x1+clockCenterX, y1+clockCenterY,
x2+clockCenterX, y2+clockCenterY);
}

void drawSec(int s)
{
    float x1, y1, x2, y2;

    s=s*6;
    s=s+270;

    x1=29*cos(s*0.0175);
    y1=29*sin(s*0.0175);
    x2=26*cos(s*0.0175);
    y2=26*sin(s*0.0175);

    if ((s % 5) == 0)
myOLED.clrLine(x1+clockCenterX, y1+clockCenterY,
x2+clockCenterX, y2+clockCenterY);
    else
myOLED.drawLine(x1+clockCenterX, y1+clockCenterY,
x2+clockCenterX, y2+clockCenterY);
}

void drawMin(int m)
{
    float x1, y1, x2, y2, x3, y3, x4, y4;

    m=m*6;
    m=m+270;

    x1=25*cos(m*0.0175);
    y1=25*sin(m*0.0175);
    x2=3*cos(m*0.0175);
    y2=3*sin(m*0.0175);
    x3=10*cos((m+8)*0.0175);
    y3=10*sin((m+8)*0.0175);
    x4=10*cos((m-8)*0.0175);
    y4=10*sin((m-8)*0.0175);

myOLED.drawLine(x1+clockCenterX, y1+clockCenterY,
x3+clockCenterX, y3+clockCenterY);
myOLED.drawLine(x3+clockCenterX, y3+clockCenterY,
x2+clockCenterX, y2+clockCenterY);
myOLED.drawLine(x2+clockCenterX, y2+clockCenterY,
x4+clockCenterX, y4+clockCenterY);
myOLED.drawLine(x4+clockCenterX, y4+clockCenterY,
x1+clockCenterX, y1+clockCenterY);
}

void drawHour(int h, int m)
{
    float x1, y1, x2, y2, x3, y3, x4, y4;

    h=(h*30)+(m/2);
    h=h+270;

```

```

x1=20*cos(h*0.0175);
y1=20*sin(h*0.0175);
x2=3*cos(h*0.0175);
y2=3*sin(h*0.0175);
x3=8*cos((h+12)*0.0175);
y3=8*sin((h+12)*0.0175);
x4=8*cos((h-12)*0.0175);
y4=8*sin((h-12)*0.0175);

myOLED.drawLine(x1+clockCenterX, y1+clockCenterY,
x3+clockCenterX, y3+clockCenterY);
myOLED.drawLine(x3+clockCenterX, y3+clockCenterY,
x2+clockCenterX, y2+clockCenterY);
myOLED.drawLine(x2+clockCenterX, y2+clockCenterY,
x4+clockCenterX, y4+clockCenterY);
myOLED.drawLine(x4+clockCenterX, y4+clockCenterY,
x1+clockCenterX, y1+clockCenterY);
}

void printDate()
{
    Time t_temp;

    t_temp = rtc.getTime();
    myOLED.print(rtc.getDOWStr(FORMAT_SHORT), RIGHT, 0);
    if (t_temp.date<10)
        myOLED.printNumI(t_temp.date, 96, 16);
    else
        myOLED.printNumI(t_temp.date, 88, 16);
    myOLED.print(rtc.getMonthStr(FORMAT_SHORT), RIGHT, 32);
    myOLED.printNumI(t_temp.year, RIGHT, 47);
}

void loop()
{
    int x, y;
    int prevSec;

    drawDisplay();
    drawSec(t.sec);
    drawMin(t.min);
    drawHour(t.hour, t.min);
    printDate();
    myOLED.update();

    prevSec = t.sec;
    while (t.sec == prevSec)
    {
        if (Serial.available()>0)
            serialMode();
        delay(100);
        t = rtc.getTime();
    }
}

```

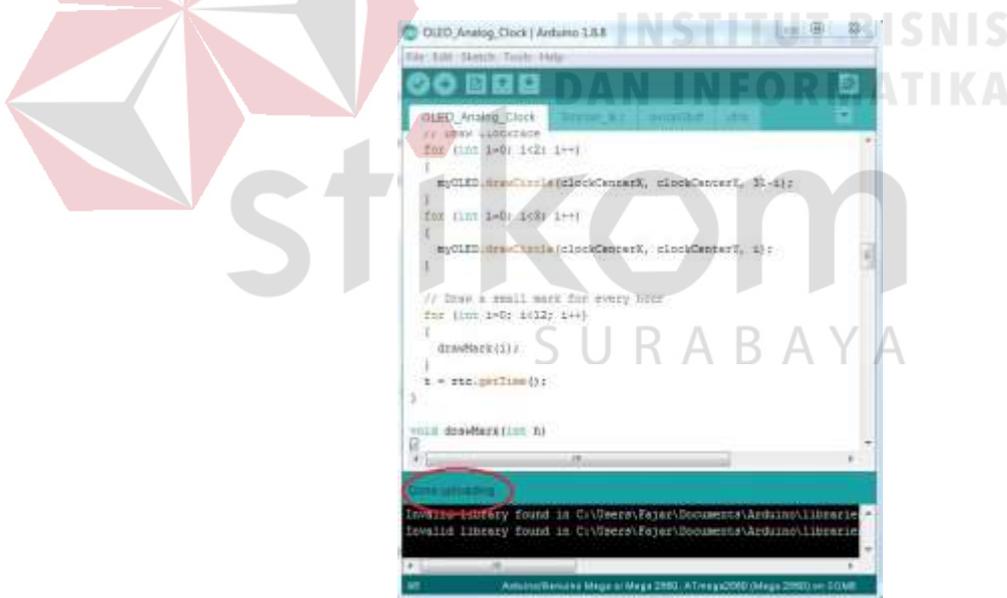
- d. Apabila telah selesai untuk mengisi perintah, maka tekan “Verify” untuk mengecek apabila terdapat perintah yang salah dalam bahasa C. Dan

tekan “*Upload*” untuk memasukkan perintah tersebut ke dalam Arduino Uno.

- e. Setelah *program* telah berhasil dimasukkan, maka tekan *icon Serial monitor* pada kanan atas. Maka akan muncul tampilan serial monitor.
- f. Setelah window serial monitor muncul, amati kiriman data serial oleh Arduino.

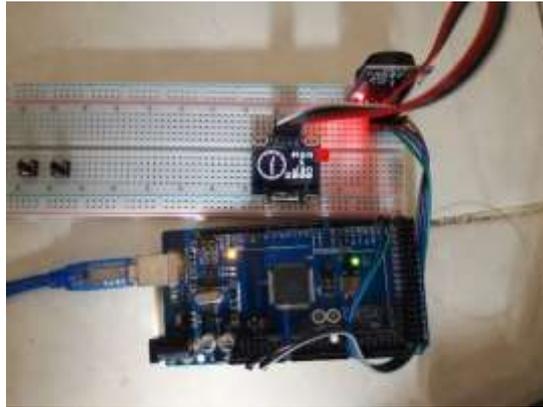
#### 4.3.4 Hasil Pengujian

Hasil dari pengujian pengisian program ke Arduino dapat dilihat pada Gambar 4.5. Lingkaran merah menunjukkan bahwa Arduino yang digunakan berhasil di isi dengan program yang telah ditulis dalam software Arduino IDE.



**Gambar 4.5** *upload* program berhasil

Program yang dimasukkan kedalam Arduino merupakan program sederhana untuk mencoba apakah RTC dapat berjalan atau tidak yang terdapat pada Gambar 4.5.



**Gambar 4.6** Program Berhasil Berjalan

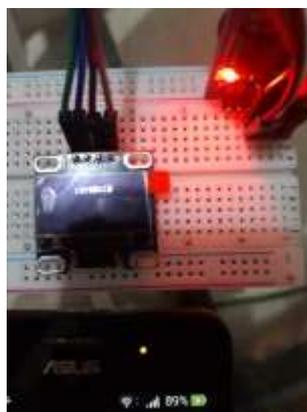
Gambar 4.6 menunjukkan bahwa *program* dapat berjalan dan menampilkan sesuai yang di inginkan oleh pengguna pada RTC.

#### 4.4. Pengujian Sistem

Disini akan ditunjukkan mengenai hasil pengujian terhadap sistem

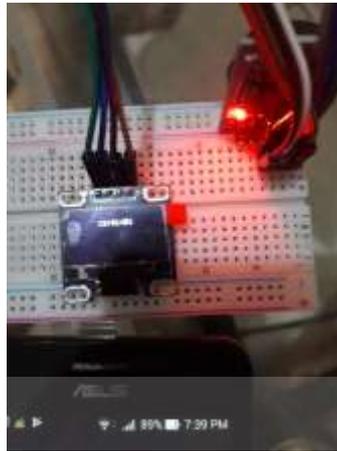
##### 4.4.1 Pengujian Mode Pertama

1. Pengujian Pengujian pertama dilakukan pada pengisian baterai *handphone* selama satu menit. Percobaan dapat di lihat pada Gambar 4.7



**Gambar 4.7** Posisi Pertama Alat di Nyalakan

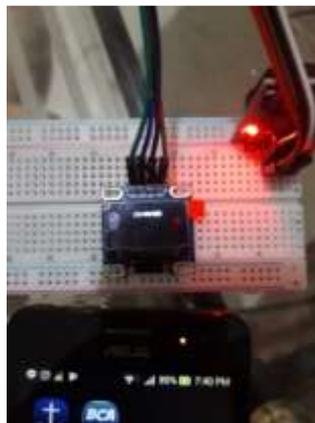
Setelah alat terhubung maka dapat dilihat pengisian baterai sedang berjalan dan layar LCD OLED menampilkan waktu yang berjalan. Lalu setelah dilakukan selama satu menit maka di dapat hasil seperti pada Gambar 4.8



**Gambar 4.8** Hasil Percobaan Setelah Satu Menit

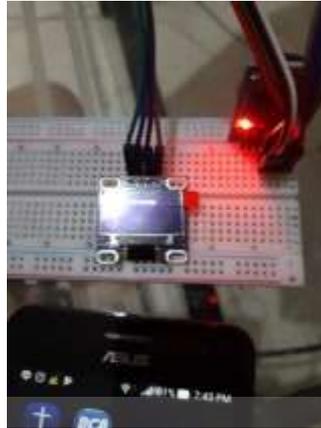
Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada *handphone* namun presentase baterai *handphone* tidak bertambah karena pemberian waktu yang sebentar.

2. Pengujian Kedua dilakukan pada pengisian baterai *handphone* selama tiga menit. Percobaan dapat di lihat pada Gambar 4.9



**Gambar 4.9** Posisi Pertama Alat di Nyalakan

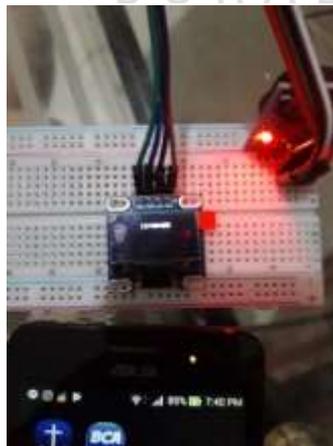
Setelah alat terhubung maka dapat dilihat pengisian baterai sedang berjalan dan layar LCD OLED menampilkan waktu yang berjalan. Lalu setelah dilakukan selama tiga menit maka di dapat hasil seperti pada Gambar 4.10



**Gambar 4.10** Hasil Percobaan Setelah Tiga Menit

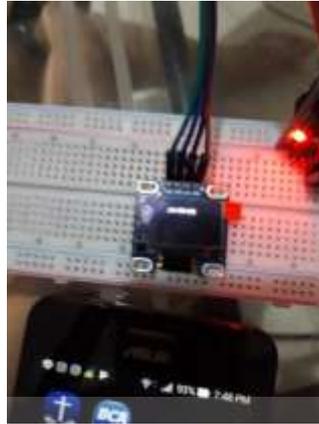
Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada handphone dan presentase baterai handphone bertambah.

3. Pengujian ketiga dilakukan pada pengisian baterai *handphone* selama lima menit. Percobaan dapat di lihat pada Gambar 4.11



**Gambar 4.11** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat pengisian baterai sedang berjalan dan layar LCD OLED menampilkan waktu yang berjalan. Lalu setelah dilakukan selama lima menit maka di dapat hasil seperti pada Gambar 4.12



**Gambar 4.12** Hasil Percobaan Setelah Lima Menit

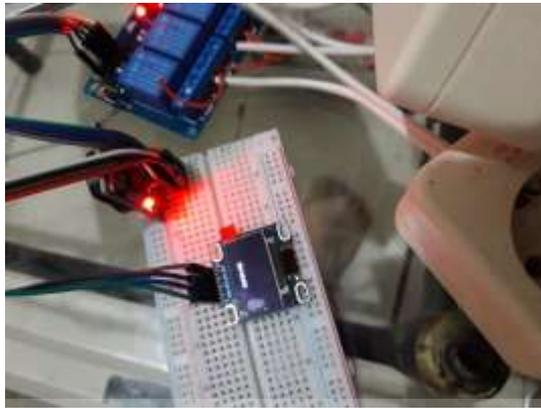
Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada handphone dan presentase baterai handphone bertambah.

4. Pengujian pertama dilakukan pada lampu yang di nyalakan selama satu menit. Percobaan dapat di lihat pada Gambar 4.13



**Gambar 4.13** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah dilakukan selama satu menit maka di dapat hasil seperti pada Gambar 4.14



**Gambar 4.14** Hasil Percobaan Setelah Satu Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada lampu dan lampu padam.

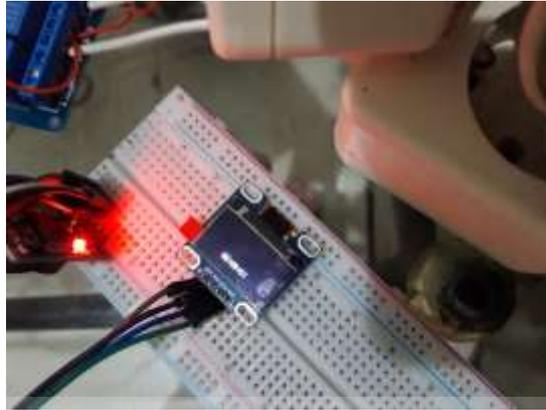
5. Pengujian kedua dilakukan pada lampu yang di nyalakan selama satu menit.

Percobaan dapat di lihat pada Gambar 4.15



**Gambar 4.15** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah dilakukan selama tiga menit maka di dapat hasil seperti pada Gambar 4.16

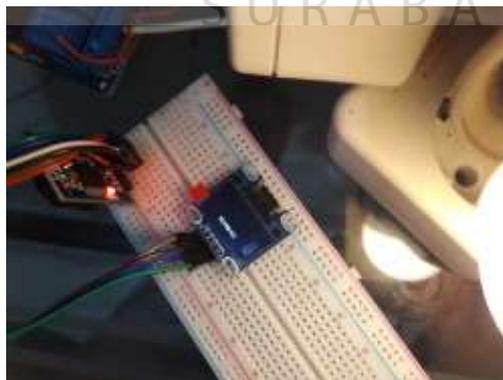


**Gambar 4.16** Hasil Percobaan Setelah Tiga Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada lampu dan lampu padam.

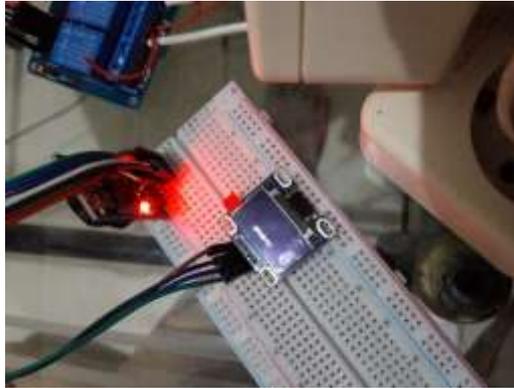
6. Pengujian ketiga dilakukan pada lampu yang di nyalakan selama lima menit.

Percobaan dapat di lihat pada Gambar 4.17



**Gambar 4.17** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah dilakukan selama lima menit maka di dapat hasil seperti pada Gambar 4.18



**Gambar 4.18** Hasil Percobaan Setelah Lima Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada lampu dan lampu padam.

7. Pengujian pertama dilakukan pada pengisian baterai laptop selama satu menit. Percobaan dapat di lihat pada Gambar 4.19



**Gambar 4.19** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama satu menit maka di dapat hasil seperti pada Gambar 4.20



**Gambar 4.20** Hasil Percobaan Setelah Satu Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada laptop dan presentase baterai laptop bertambah.

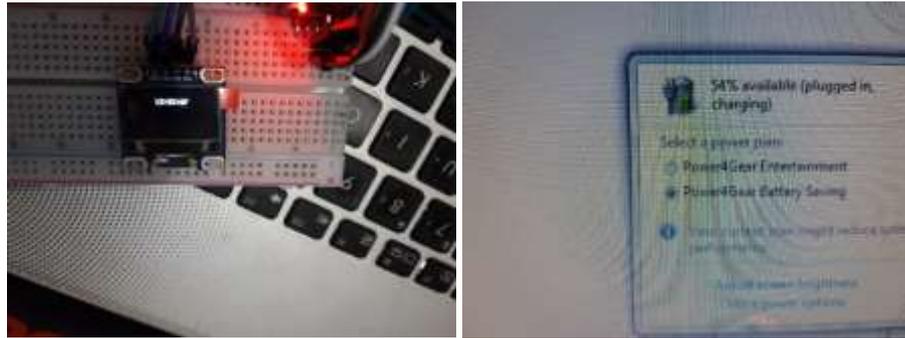
8. Pengujian kedua dilakukan pada pengisian baterai laptop selama tiga menit.

Percobaan dapat di lihat pada Gambar 4.21



**Gambar 4.21** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama tiga menit maka di dapat hasil seperti pada Gambar 4.22



**Gambar 4.22** Hasil Percobaan Setelah Tiga Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada laptop dan presentase baterai laptop bertambah.

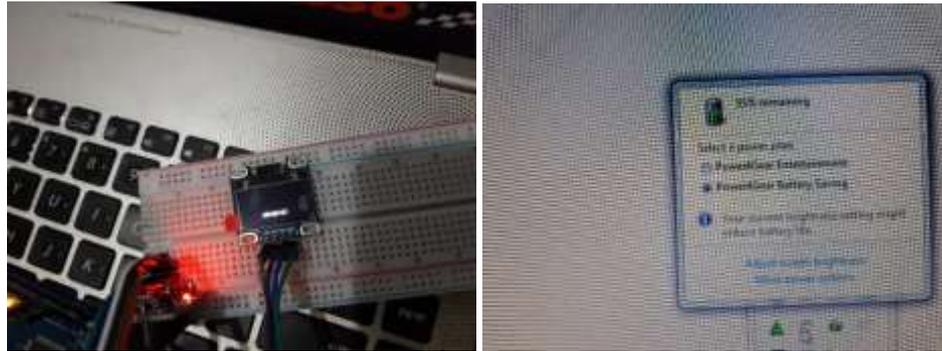
9. Pengujian ketiga dilakukan pada pengisian baterai laptop selama lima menit.

Percobaan dapat di lihat pada Gambar 4.23



**Gambar 4.23** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layar LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama lima menit maka di dapat hasil seperti pada Gambar 4.24



**Gambar 4.24** Hasil Percobaan Setelah Lima Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada laptop dan presentase baterai laptop bertambah.

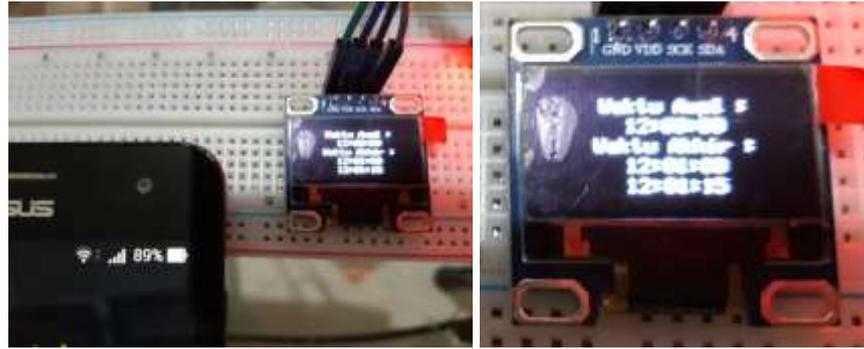
#### 4.4.2 Pengujian Mode Kedua

1. Pengujian pertama dilakukan pada pengisian baterai *handphone* selama satu menit, mulai jam 12.00.00 sampai jam 12.01.00. Percobaan dapat di lihat pada Gambar 4.25



**Gambar 4.25** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat pengisian baterai sedang berjalan dan layar LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama satu menit maka di dapat hasil seperti pada Gambar 4.26



**Gambar 4.26** Hasil Percobaan Setelah Satu Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada handphone namun presentase baterai handphone tidak bertambah karena pemberian waktu yang sebentar.

2. Pengujian kedua dilakukan pada pengisian baterai *handphone* selama tiga menit, mulai jam 12.00.00 sampai jam 12.03.00 Percobaan dapat di lihat pada Gambar 4.27



**Gambar 4.27** Posisi Pertama Alat di Nyalakan

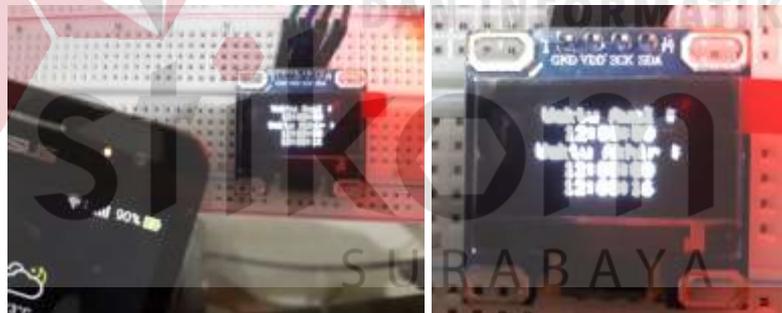
Setelah alat terhubung maka dapat dilihat pengisian baterai sedang berjalan dan layar LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama tiga menit maka di dapat hasil seperti pada Gambar 4.28



**Gambar 4.28** Hasil Percobaan Setelah Tiga Menit

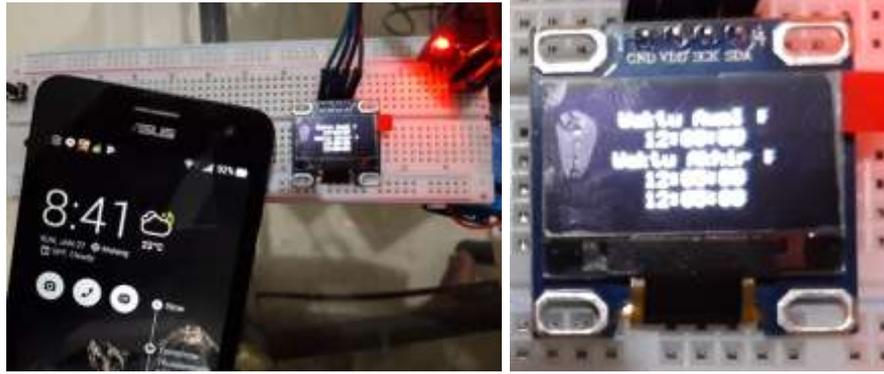
Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada handphone dan presentase baterai handphone bertambah.

3. Pengujian ketiga dilakukan pada pengisian baterai *handphone* selama lima menit, mulai jam 12.00.00 sampai jam 12.05.00 Percobaan dapat di lihat pada Gambar 4.29



**Gambar 4.29** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat pengisian baterai sedang berjalan dan layar LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama lima menit maka di dapat hasil seperti pada Gambar 4.30



**Gambar 4.30** Hasil Percobaan Setelah Lima Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada handphone dan presentase baterai handphone bertambah.

4. Pengujian pertama dilakukan pada lampu yang di nyalakan selama satu menit, mulai jam 12.00.00 sampai jam 12.01.00 Percobaan dapat di lihat pada Gambar 4.31



**Gambar 4.31** Posisi Pertama Alat di Nyalakan

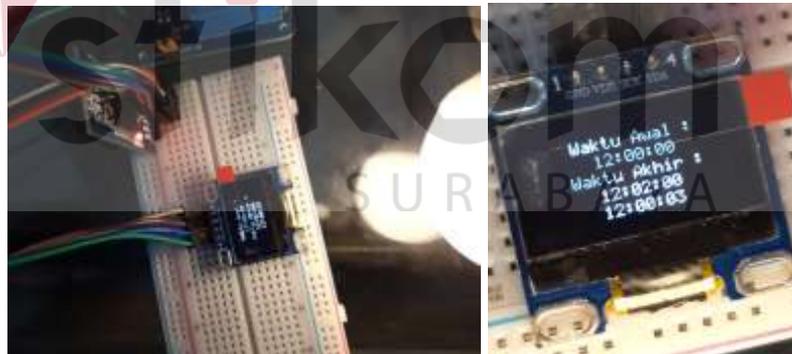
Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama satu menit maka di dapat hasil seperti pada Gambar 4.32



**Gambar 4.32** Hasil Percobaan Setelah Satu Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada lampu dan lampu padam.

5. Pengujian kedua dilakukan pada lampu yang di nyalakan selama dua menit, mulai jam 12.00.00 sampai jam 12.02.00. Percobaan dapat di lihat pada Gambar 4.33



**Gambar 4.33** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama dua menit maka di dapat hasil seperti pada Gambar 4.34



**Gambar 4.34** Hasil Percobaan Setelah Dua Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada lampu dan lampu padam.

6. Pengujian ketiga dilakukan pada lampu yang di nyalakan selama tiga menit, mulai jam 12.00.00 sampai jam 12.03.00 Percobaan dapat di lihat pada Gambar 4.35



**Gambar 4.35** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama tiga menit maka di dapat hasil seperti pada Gambar 4.36



**Gambar 4.36** Hasil Percobaan Setelah Tiga Menit

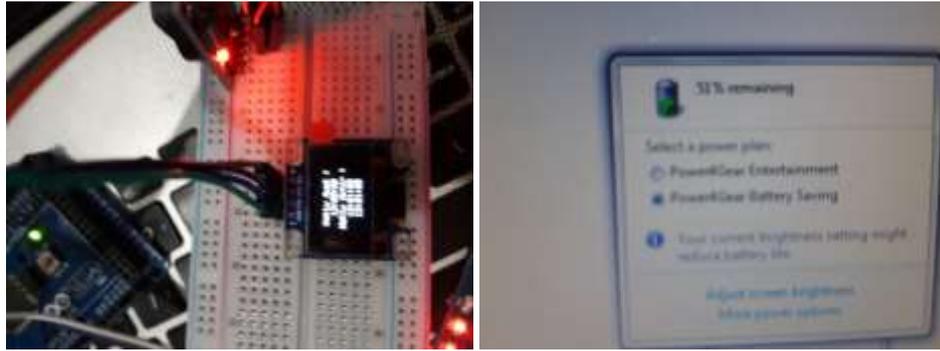
Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada lampu dan lampu padam.

7. Pengujian pertama dilakukan pada pengisian baterai laptop selama satu menit, mulai jam 12.00.00 sampai jam 12.01.00 Percobaan dapat di lihat pada Gambar 4.37



**Gambar 4.37** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama satu menit maka di dapat hasil seperti pada Gambar 4.38



**Gambar 4.38** Hasil Percobaan Setelah Satu Menit

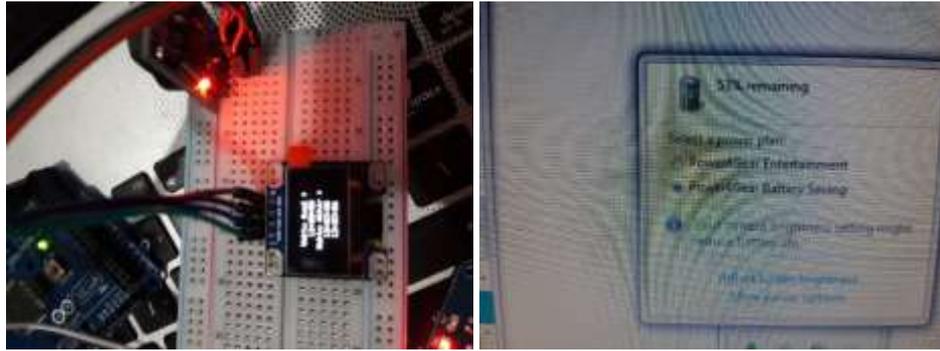
Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada laptop dan presentase baterai laptop bertambah.

8. Pengujian kedua dilakukan pada pengisian baterai laptop selama tiga menit, mulai jam 12.00.00 sampai jam 12.03.00. Percobaan dapat dilihat pada Gambar 4.39



**Gambar 4.39** Posisi Pertama Alat di Nyalakan

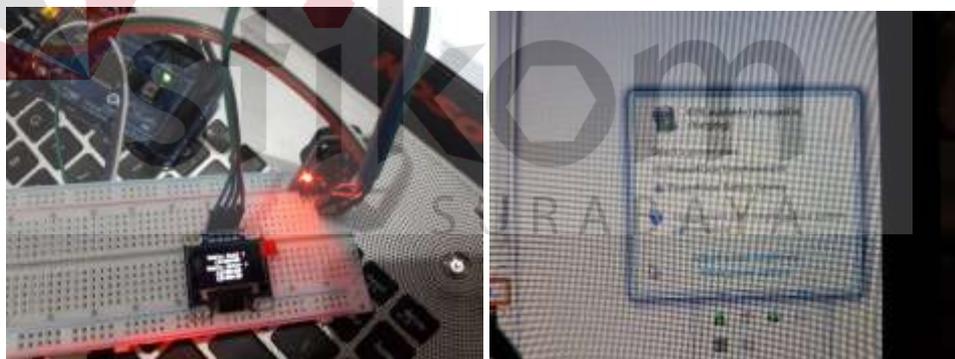
Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah dilakukan selama tiga menit maka di dapat hasil seperti pada Gambar 4.40



**Gambar 4.40** Hasil Percobaan Setelah Tiga Menit

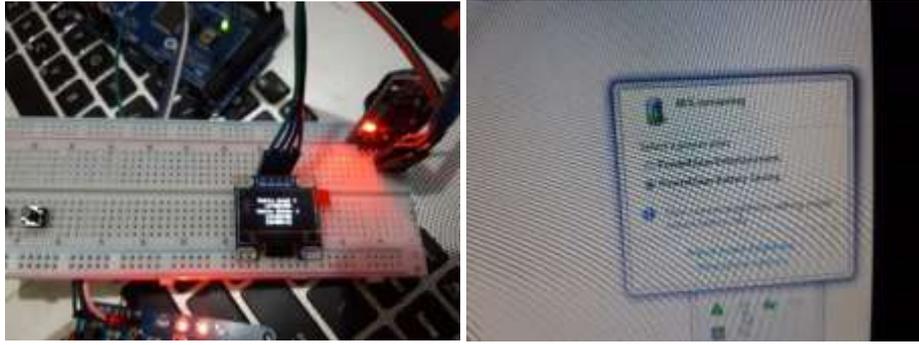
Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada laptop dan presentase baterai laptop bertambah.

9. Pengujian ketiga dilakukan pada pengisian baterai laptop selama lima menit, mulai jam 12.00.00 sampai jam 12.05.00. Percobaan dapat di lihat pada Gambar 4.41



**Gambar 4.41** Posisi Pertama Alat di Nyalakan

Setelah alat terhubung maka dapat dilihat jika lampu menyala dan layer LCD OLED menampilkan waktu yang berjalan. Lalu setelah di lakukan selama lima menit maka di dapat hasil seperti pada Gambar 4.42



**Gambar 4.42** Hasil Percobaan Setelah Lima Menit

Dari situ dapat dilihat bahwa *Relay* berhasil memutus aliran listrik kepada laptop dan presentase baterai laptop bertambah.

#### 4.4.3 Penjelasan Hasil Uji Coba

##### 1. Hasil Uji Mode 1

Tabel 4.1 sampai Tabel 4.3 menjelaskan tentang hasil pengujian yang dilakukan pada mode 1. Dimana mode ke 1 adalah pilihan menu pertama yang menjelaskan tentang penginputan waktu dimana waktu ini sendiri berfungsi sebagai alat ukur berapa lama alat akan digunakan oleh pengguna. Dengan memberi inputan waktu, pengguna dapat menggunakan alat sesuai kebutuhan.

Contoh jika pengguna ingin menggunakan alat selama satu menit, maka pengguna tinggal memasukkan nilai waktu selama satu menit atau sesuai yang diinginkan. Lalu alat akan bekerja sesuai waktu yang telah diberikan oleh pengguna.

Contoh keberhasilan mode ke 1 ini dapat dilihat pada Tabel 4.1 sampai Tabel 4.3. Dimana setelah dilakukan percobaan sebanyak 9 kali dan diberi waktu

yang berbeda-beda alat dapat bekerja sesuai inputan dan dapat memutuskan arus listrik.

**Tabel 4.1** Tabel Percobaan Saklar 1 Mode 1

No.	Saklar	Durasi	Hasil
1	1	1 Menit	Berhasil menyalakan charger <i>handphone</i> selama satu menit dan memutus aliran listrik
2	1	3 Menit	Berhasil menyalakan charger <i>handphone</i> selama tiga menit dan memutus aliran listrik
3	1	5 Menit	Berhasil menyalakan charger <i>handphone</i> selama lima menit dan memutus aliran listrik

**Tabel 4.2** Tabel Percobaan Saklar 2 Mode 1

No.	Saklar	Durasi	Hasil
1	2	2 Menit	Berhasil menyalakan charger <i>handphone</i> dan memutus aliran listrik serta prosentase daya baterai bertambah
2	2	4 Menit	Berhasil menyalakan charger <i>handphone</i> dan memutus aliran listrik serta prosentase daya baterai bertambah
3	2	6 Menit	Berhasil menyalakan charger <i>handphone</i> dan memutus aliran listrik serta prosentase daya baterai bertambah

**Tabel 4.3** Tabel Percobaan Saklar 3 Mode 1

No.	Saklar	Durasi	Hasil
1	3	2 Menit	Berhasil menyalakan charger <i>handphone</i> selama dua menit dan memutus aliran listrik
2	3	5 Menit	Berhasil menyalakan charger <i>handphone</i> selama lima menit dan memutus aliran listrik
3	3	7 Menit	Berhasil menyalakan charger <i>handphone</i> selama tujuh menit dan memutus aliran listrik

## 2. Hasil Uji Mode 2

**Tabel 4.4** Tabel Percobaan Saklar 1 Mode 2

No.	Saklar	Durasi	Hasil
1	1	12.00.00 – 12.01.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00–12.01.00 dan memutus aliran listrik
2	1	12.00.00 – 12.03.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00–12.03.00 dan memutus aliran listrik
3	1	12.00.00 – 12.05.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00–12.05.00 dan memutus aliran listrik

**Tabel 4.5** Tabel Percobaan Saklar 2 Mode 2

No.	Saklar	Durasi	Hasil
1	2	12.00.00 – 12.02.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00–12.02.00 dan memutus aliran listrik serta prosentase daya baterai bertambah
2	2	12.00.00 – 12.04.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00–12.04.00 dan memutus aliran listrik serta prosentase daya baterai bertambah
3	2	12.00.00 – 12.06.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00–12.06.00 dan memutus aliran listrik serta prosentase daya baterai bertambah

**Tabel 4.6** Tabel Percobaan Saklar 3 Mode 2

No.	Saklar	Durasi	Hasil
1	3	12.00.00 – 12.02.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00 – 12.02.00 dan memutus aliran listrik
2	3	12.00.00 – 12.05.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00 – 12.05.00 dan memutus aliran listrik serta prosentase daya baterai bertambah
3	3	12.00.00 – 12.07.00	Berhasil menyalakan charger <i>handphone</i> mulai pukul 12.00.00 – 12.07.00 dan memutus aliran listrik serta prosentase daya baterai bertambah

Tabel 4.4 sampai Tabel 4.6 menjelaskan tentang hasil pengujian yang dilakukan pada mode 2. Dimana mode ke 2 adalah pilihan menu kedua yang menjelaskan tentang penginputan waktu dimana pemberian waktu ini dilakukan sebanyak dua kali yaitu pemberian waktu awal dan waktu akhir. Waktu ini sendiri berfungsi sebagai alat ukur berapa lama alat akan digunakan oleh pengguna.

Contoh jika pengguna ingin menggunakan alat selama satu menit, maka pengguna memberi inputan nilai awal dan nilai akhir. Contoh nilai awal 00.00.00 dan nilai akhir 00.01.00. maka alat bekerja selama satu menit sesuai inputan yang diberikan.

Contoh keberhasilan mode ke 2 ini dapat dilihat pada Tabel 4.4 sampai Tabel 4.6. Dimana setelah dilakukan percobaan sebanyak 9 kali dan diberi waktu berbeda-beda alat dapat bekerja sesuai inputan dan dapat memutuskan arus listrik.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

1. Setelah dilakukan percobaan sebanyak 9 kali dengan waktu yang berbeda-beda untuk memutuskan arus listrik berdasarkan *setting* mode 1 sebagaimana pada Tabel 4.1 sampai Tabel 4.3 maka didapatkan tingkat keberhasilan alat sebesar 100%.
2. Berdasarkan *setting* mode 2 sebagaimana pada Tabel 4.4 sampai Tabel 4.6 dimana telah dilakukan percobaan sebanyak 9 kali dengan waktu yang berbeda-beda untuk menyalakan dan memutuskan arus listrik mempunyai tingkat keberhasilan alat sebesar 100%.

#### 5.2 Saran

1. Disarankan untuk pengembangan hasil tugas akhir ini dapat dilakukan *monitoring* dan kontroling secara jarak jauh menggunakan *smartphone*.
2. Fitur alat tugas akhir ini dapat mendukung *setting* dan konfigurasi secara *multi-timer*, sehingga dapat menyalakan dan memutuskan arus listrik beberapa kali dalam sehari.

## DAFTAR PUSTAKA

- Asep Kurniawan. 2018. Seting waktu RTC otomatis dengan Arduino di <https://www.semesin.com/project/2018/05/09/seting-waktu-rtc-otomatis-dengan-Arduino/> di akses pada 28 September 2018
- Fatimah, D. D. (2017). Perancangan Pengendali Lampu Rumah Otomatis Berbasis Arduino Nano. *Jurnal A;goritma*, 14, 352-359.
- Jimmi Sitepu. 2018. Fungsi dan Pengertian Arduino, Manfaat dan Cara Kerjanya di <https://mikroavr.com/fungsi-dan-cara-kerja-Arduino/> di akses pada 26 September 2018
- Kusuma Wardana. 2016. [TUTORIAL] Menggunakan Real Time Clock (RTC) pada Arduino di <https://tutorkeren.com/artikel/tutorial-menggunakan-real-time-clock-rtc-pada-Arduino.htm> di akses pada 26 September 2018
- Novalius, F. (2018, Februari 17). Indonesia Pengguna Smartphone Ke-4 Dunia, Begini Tekad Menperin Dongkrak Industri Telematika. Dipetik September 27, 2018, dari OkezoneFinance: <https://economy.okezone.com/read/2018/02/17/320/1860752/>
- Prastiantari, A. (2017). *SKOPIN (STOP KONTAK PINTAR) PENGENDALI ARUS LISTRIK MENGGUNAKAN TIMER PADA STOP KONTAK BERBASIS ARDUINO*. *JKOMA*, 1, 21-28.