



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 145561**

**SISTEM KOMUNIKASI DAN MONITORING ADAPTIF RELAI  
ARUS LEBIH TERHADAP SUMBER TEGANGAN PADA  
SALURAN LISTRIK SATU FASA**

Nanda Lintang Syafitri  
NRP 2214038006

Dosen Pembimbing  
Dr. Eng. Ardyono Priyadi, S.T., M.Eng.

PROGRAM STUDI TEKNIK LISTRIK  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE 145561**

**ADAPTIVE OVERCURRENT RELAY COMMUNICATION AND  
MONITORING SYSTEM TO VOLTAGE SOURCE ON ONE PHASE  
ELECTRIC CHANNEL**

Nanda Lintang Syafitri  
NRP 2214038006

Advisor  
Dr. Eng. Ardyono Priyadi, S.T., M.Eng.

ELECTRICAL ENGINEERING STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Sistem Komunikasi dan Monitoring Adaptif Relai Arus Lebih Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Juli 2017



Nanda Lintang Syafitri  
NRP 2214038006

-----Halaman ini sengaja dikosongkan-----

**SISTEM KOMUNIKASI DAN MONITORING ADAPTIF RELAI  
ARUS LEBIH TERHADAP SUMBER TEGANGAN PADA  
SALURAN LISTRIK SATU FASA**

**TUGAS AKHIR**

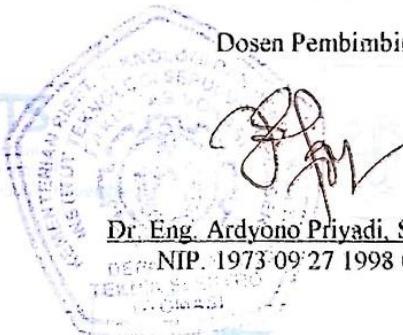
Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya

Pada

Program Studi Teknik Listrik  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing



Dr. Eng. Ardyono Priyadi, S.T., M.Eng.

NIP. 1973-09-27 1998 03 1004

**SURABAYA**

**JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

# SISTEM KOMUNIKASI DAN MONITORING ADAPTIF RELAI ARUS LEBIH TERHADAP SUMBER TEGANGAN PADA SALURAN LISTRIK SATU FASA

**Nama** : Nanda Lintang Syafitri  
**NRP** : 2214038006  
**Pembimbing** : Dr. Eng. Ardyono Priyadi, S.T., M.Eng.  
**NIP** : 1973 09 27 1998 03 1004

## ABSTRAK

PT PLN (Persero) mengalami peningkatan jumlah pelanggan rata-rata 8,4% per tahun. Pertambahan pelanggan dapat diindikasikan sebagai pertambahan beban. Agar pelanggan atau beban dapat terus mendapatkan suplai listrik, pertambahan beban ini juga harus diiringi dengan pertambahan sumber. Karena sumber tidak akan mampu mensuplai jika pertambahan beban terus menerus tanpa adanya pertambahan sumber yang lain.

Permasalahannya adalah, jika semakin besar beban dan sumber yang ada, maka arus nominal dan arus gangguan juga akan semakin besar. Sedangkan jika arus yang ada terlalu besar, dapat menyebabkan kerusakan pada peralatan yang lain. Sehingga dibutuhkan pengaman atau relai yang *setting* arusnya dapat berubah sesuai dengan sumber tegangan yang ada.

Pada Tugas Akhir ini dapat mendeteksi berapa jumlah sumber yang mensuplai jaringan, dan selanjutnya disebut relai indikator. Selain dapat mendeteksi jumlah sumber, relai juga dapat berkomunikasi satu sama lain, sehingga memungkinkan pengiriman beberapa data antar relai. Ketika relai indikator mendeteksi jumlah sumber, data sumber akan dikirimkan ke relai utama. Relai utama ini bersifat adaptif, yaitu *setting* arusnya dapat berubah sesuai dengan jumlah sumber. Adaptif relai ini yang nantinya bertugas untuk koordinasi pengamanan jaringan. Pada monitoring adaptif relai dapat menunjukkan perubahan yang terjadi pada sumber, arus dan tegangan. Dan monitor ini memiliki delay pembacaan data data 2 detik. Dan koordinasi relai didapatkan nilai error waktu trip untuk sumber 1 adalah 0,022% pada relai 1 dan 0,048% pada relai 2. Dan ketika 2 sumber memiliki error waktu trip sebesar 0,038% pada relai 1 dan 0,063% pada relai 2.

**Kata Kunci** : *Over Current Relay*, Adaptif Relai, Komunikasi *Master Slave*, Relai Indikator, *Interface LabVIEW*

-----Halaman ini sengaja dikosongkan-----

***ADAPTIVE OVERCURRENT RELAY COMMUNICATION AND  
MONITORING SYSTEM TO VOLTAGE SOURCE ON ONE PHASE  
ELECTRIC CHANNEL***

***Name*** : Nanda Lintang Syafitri  
***NRP*** : 2214038006  
***Advisor*** : Dr. Eng. Ardyono Priyadi, S.T., M.Eng.  
***NIP*** : 1973 09 27 1998 03 1004

***ABSTRACT***

*PT PLN (Persero) has increased the number of customers an average of 8.4% for each year. Increased subscribers may be indicated as an increase in expenses. In order for customers or loads to continue to get electricity supply, this burden must also be accompanied by additional sources. Because the source will not be able to supply if the load increases continuously without any increase in other sources.*

*The problem is, if the greater the load and the available source, the nominal current and the noise current will also be greater. Whereas if the current is too large, it can cause damage to other equipment. So it takes a safety or relays that the current setting can change according to the existing voltage source.*

*In this Final Project can detect how many sources that supply network, and hereinafter referred as indicator relay. In addition to being able to detect the number of sources, relays can also communicate with each other, thus allowing the transmission of some data between relays. When the indicator relay detects the number of sources, the source data will be sent to the main relay. This main relay is adaptive, ie the current setting can change according to the number of sources. This adaptive relay which later served to coordinate network security. In adaptive monitoring the relays can show changes occurring at source, current and voltage. And this monitor has a data readout delay of 2 seconds. And the relay coordination of the trip error value for source 1 is 0.022% in relays 1 and 0.048% in relay 2. And when 2 sources have trip time error of 0.038% in relays 1 and 0.063% in relay 2.*

***Keywords*** : *Over Current Relay, Adaptive Relay, Indicator Relay, Master Slave Communication, LabVIEW Interfacing*

-----Halaman ini sengaja dikosongkan-----

## KATA PENGANTAR

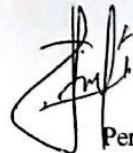
Dengan mengucap puji kami panjatkan kehadirat Allah Swt yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Adapun judul Tugas Akhir ini yaitu "Sistem Komunikasi Dan Monitoring Adaptif Relai Arus Lebih Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa".

Tujuan penulisan Tugas Akhir ini dibuat sebagai salah satu syarat memperoleh gelar Ahli Madya (A. Md.) pada Program Studi Teknik Listrik, Departemen Teknik Elektro Otomasi, Institut Teknologi Sepuluh Nopember. Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, tugas akhir ini tidak dapat terselesaikan dengan baik. Oleh karenanya penulis ingin menyampaikan ucapan terimakasih kepada :

1. Kepala Departemen Teknik Elektro Otomasi
2. Bapak Dr. Eng. Ardyono Priyadi, S.T., M.Eng. selaku dosen pembimbing Tugas Akhir.
3. Staff/Karyawan/Dosen Departemen Teknik Elektro Otomasi.
4. Orang tua yang saya sayangi serta adik-adik dan juga sahabat dekat yang telah memberikan dukungan moral dan spiritual.
5. Rekan-rekan seperjuangan mahasiswa Departemen Teknik Elektro Otomasi angkatan 2014.

Penulis menyadari sepenuhnya bahwa tugas akhir ini jauh dari kata sempurna, oleh karena itu penulis mengharapkan saran dan kritik yang bersifat membangun untuk memperbaiki kekurangan dalam Tugas Akhir ini. Akhir kata penulis berharap semoga Tugas Akhir ini berguna bagi penulis pada khususnya dan bagi pembaca umumnya

Surabaya, 19 Juli 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

HALAMAN

HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR .....	v
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xix
BAB I.....	1
1.1    Latar Belakang.....	1
1.2    Permasalahan .....	2
1.3    Batasan Masalah .....	2
1.4    Tujuan.....	2
1.5    Metodologi Penelitian.....	2
1.6    Sistematika Laporan .....	3
1.7    Relevansi.....	4
BAB II.....	5
2.1    Arduino UNO .....	5
2.2    Arduino Mega 2560 .....	6
2.3    Arduino IDE .....	7
2.4    Modul Komunikasi RS485 .....	8
2.5 <i>Relay</i> AZ943.....	9
2.6 <i>Relay</i> AC MY2 220 Volt .....	9
2.7 <i>Real Time Clock</i> DS1307 <i>modules</i> (RTC) .....	10
2.8    Modul <i>SD Card</i> / MMC.....	11

2.9	<i>Miniature Circuit Breaker (MCB)</i> .....	13
2.10	<i>Uninterruptible Power Supply (UPS)</i> .....	13
2.11	<i>Liquid Crystal Display Keypad (LCD Keypad)</i> .....	14
2.12	<i>Software LabVIEW</i> .....	16
<b>BAB III</b>	.....	<b>21</b>
3.1	Diagram Fungsional Alat.....	21
3.2	Perancangan Mekanik .....	23
3.3	Perancangan Elektronik .....	24
3.3.1	Perancangan Relai Indikator .....	25
3.3.2	Perancangan Komunikasi RS485 .....	26
3.3.3	Perancangan LCD Keypad .....	27
3.3.4	Perancangan RTC DS1307 dan <i>Data Logger</i> .....	28
3.4	Perancangan Perangkat Lunak ( <i>Software</i> ).....	29
3.4.1	Pemrograman <i>Software</i> Arduino IDE .....	30
3.4.2	Pemrograman <i>Software</i> LabView .....	37
<b>BAB IV</b>	.....	<b>41</b>
4.1	<i>Input / Output</i> Arduino .....	41
4.2	Pengujian Relai Indikator .....	45
4.3	Pembacaan RTC .....	48
4.4	Memori SD Card ( <i>Data Logger</i> ).....	51
4.5	Pengujian Komunikasi RS485.....	56
4.6	Tampilan LCD Keypad 16x2.....	59
4.7	Pengujian <i>Software</i> LabVIEW .....	61
4.8	Pengujian Koordinasi Adaptif Relai .....	65
4.9	Analisa Relevansi .....	68
<b>BAB V</b>	.....	<b>69</b>
5.1	Kesimpulan.....	69

5.2	Saran .....	69
	DAFTAR PUSTAKA .....	71
	LAMPIRAN-A .....	A-1
	LAMPIRAN-B .....	B-1
	LAMPIRAN-C .....	C-1
	RIWAYAT HIDUP PENULIS .....	D-1

-----Halaman ini sengaja dikosongkan-----

## DAFTAR GAMBAR

### HALAMAN

Gambar 2.1	<i>Board</i> Arduino UNO .....	5
Gambar 2.2	<i>Board</i> Arduino Mega 2560.....	6
Gambar 2.3	Jendela Arduino IDE .....	7
Gambar 2.4	Modul Komunikasi RS485 .....	8
Gambar 2.5	<i>Relay</i> AZ943.....	9
Gambar 2.6	<i>Relay</i> AC MY2 220 Volt .....	10
Gambar 2.7	RTC Tiny I2C <i>modules</i> .....	11
Gambar 2.8	Modul SD <i>Card</i> / MMC .....	12
Gambar 2.9	<i>Miniature Circuit Breaker</i> (MCB).....	13
Gambar 2.10	<i>Uninterruptible Power Supply</i> (UPS).....	14
Gambar 2.11	<i>Liquid Crystal Display Keypad</i> (LCD <i>Keypad</i> ) .....	15
Gambar 2.12	Tampilan Awal LabVIEW .....	16
Gambar 2.13	<i>Front Panel</i> .....	17
Gambar 2.14	Blok Diagram .....	18
Gambar 2.15	<i>Control Palette</i> .....	19
Gambar 2.16	<i>Functions Palette</i> .....	20
Gambar 3.1	Skema Sistem Secara Keseluruhan.....	22
Gambar 3.2	Skema Relai Indikator .....	23
Gambar 3.3	Rancangan Mekanik Relai Indikator Atas .....	23
Gambar 3.4	Rancangan Mekanik Relai Indikator Samping .....	24
Gambar 3.5	Rancangan Elektronik Relai Indikator Samping.....	24
Gambar 3.6	Rancangan Elektronik Relai Indikator Atas.....	24
Gambar 3.7	Rangkaian Resistor <i>Pull Down</i> dengan Arduino .....	26
Gambar 3.8	<i>Schematic</i> RS485 dengan Arduino Mega dan UNO ....	27
Gambar 3.9	<i>Schematic</i> LCD Keypad dengan Arduino.....	27
Gambar 3.10	<i>Schematic</i> modul RC DS1307 dengan Arduino Mega.	28
Gambar 3.11	<i>Schematic</i> microSD Adapter dengan Arduino Mega...	29
Gambar 3.12	<i>Flowchart</i> Pemrograman Keseluruhan Arduino .....	30
Gambar 3.13	<i>Flowchart</i> Pemrograman Relai Indikator .....	32
Gambar 3.14	<i>Flowchart</i> Pemrograman RTC.....	33
Gambar 3.15	<i>Flowchart</i> Pemrograman SD <i>Card</i> .....	34
Gambar 3.16	<i>Flowchart</i> Pemrograman LCD <i>Keypad</i> .....	36
Gambar 3.17	<i>Flowchart</i> Pemrograman LabVIEW.....	38
Gambar 3.18	<i>Flowchart</i> Data Serial Arduino.....	39
Gambar 4.1	Skema Pengujian Pin <i>Input</i> / <i>Output</i> Arduino .....	42
Gambar 4.2	<i>Flowchart</i> Program Pengujian Pin Arduino .....	42

Gambar 4.3	Kurva <i>Logic 1</i> Pin Arduino UNO.....	44
Gambar 4.4	Kurva <i>Logic 0</i> Pin Arduino UNO.....	45
Gambar 4.5	<i>Schematic</i> Pengujian Relai Indikator.....	46
Gambar 4.6	<i>Flowchart</i> Pengujian Relai Indikator .....	47
Gambar 4.7	<i>Flowchart</i> Pengujian RTC.....	48
Gambar 4.8	Skema Pengujian RTC .....	49
Gambar 4.9	Kurva Pembacaan RTC Relai 1.....	50
Gambar 4.10	Kurva Pembacaan RTC Relai 2.....	51
Gambar 4.11	Hasil Pengujian SD Card 1.....	52
Gambar 4.12	Hasil Pengujian SD Card 2.....	53
Gambar 4.13	<i>Flowchart</i> Penyimpanan Data .txt pada SD Card.....	54
Gambar 4.14	Tampilan LCD Penyimpanan SD Card 1 .....	55
Gambar 4.15	Tampilan LCD Penyimpanan SD Card 2 .....	56
Gambar 4.16	<i>Schematic</i> Komunikasi RS485 .....	57
Gambar 4.17	<i>Flowchart</i> Pengujian Komunikasi RS485 .....	58
Gambar 4.18	Proses Pengujian Komunikasi RS485 .....	59
Gambar 4.19	<i>Flowchart</i> Pengujian LCD .....	60
Gambar 4.20	<i>Interface</i> Sistem Monitoring Adaptif Relai .....	62
Gambar 4.21	Pengoperasian Interface .....	63
Gambar 4.22	<i>Interface</i> Membaca Overload .....	63
Gambar 4.23	Interface Membaca Overload 2 Detik .....	64
Gambar 4.24	Interface Membaca Shortcircuit .....	64
Gambar 4.25	Interface Membaca Shortcircuit 1 .....	65
Gambar 4.26	Proses Pengujian Koordinasi Adaptif Relai .....	65

## DAFTAR TABEL

HALAMAN

Tabel 2.1	Konfigurasi Pin LCD <i>Keypad</i> 16x2 .....	15
Tabel 4.1	Pengujian Relai Indikator .....	43
Tabel 4.2	Pengujian Relai Indikator .....	47
Tabel 4.3	Pengujian RTC Relai Utama 1 .....	59
Tabel 4.4	Pengujian RTC Relai Utama 2 .....	50
Tabel 4.5	Pengujian SD <i>Card</i> 1 .....	52
Tabel 4.6	Pengujian SD <i>Card</i> 2 .....	52
Tabel 4.7	Penyimpanan Data SD <i>Card</i> 1 .....	54
Tabel 4.8	Penyimpanan Data SD <i>Card</i> 2 .....	55
Tabel 4.9	Hasil Pengujian pada Komunikasi RS485.....	58
Tabel 4.10	Pengujian LCD <i>Keypad</i> Relai 1 .....	60
Tabel 4.11	Pengujian LCD <i>Keypad</i> Relai 2 .....	61
Tabel 4.12	Hasil Pengujian pada Koordinasi Relai 1 Sumber .....	66
Tabel 4.13	Hasil Pengujian pada Koordinasi Relai 2 Sumber .....	67

-----Halaman ini sengaja dikosongkan-----

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pada sebuah sistem tenaga listrikan, sistem pengamanan jaringan dan peralatan sangat penting adanya. Fungsinya untuk mengamankan dari gangguan-gangguan yang terjadi. Gangguan ini dapat disebabkan oleh kelebihan beban atau arus hubung singkat. Relai arus lebih atau *overcurrent relay* digunakan pada sistem tenaga listrik untuk mengamankan dari gangguan arus lebih tersebut. Hal ini menyebabkan relai arus lebih atau peralatan proteksi harus memiliki keandalan yang tinggi untuk menjaga keamanan pada suatu sistem. Koordinasi waktu antar relai pada saluran merupakan hal yang sangat penting pada sistem proteksi. Selain itu, relai merupakan komponen yang bekerja berdasarkan perubahan besarnya arus.

Dalam suatu sistem yang selalu berkembang, perubahan-perubahan kondisi yang terjadi akan selalu ada. Begitu juga dengan sistem kelistrikan. Menurut RUPTL PLN 2016 – 2025, kebutuhan energi listrik pada tahun 2025 akan menjadi 457 TWh, atau tumbuh rata-rata sebesar 8,6% per tahun untuk periode tahun 2016- 2025. Sedangkan beban puncak *non coincident* pada tahun 2025 akan menjadi 74.383 MW atau tumbuh rata-rata 8,4% per tahun [1].

Melihat bahwa pertumbuhan masyarakat saat ini yang sangat pesat, kebutuhan beban dari sistem kelistrikan pun akan terus meningkat. Agar beban yang semakin besar tetap dapat tersuplai aliran listrik, maka pertambahan beban ini juga harus diiringi dengan pertambahan sumber. Jika sumber bertambah maka *setting* relai yang ada juga harus berubah karena arus yang mengalir pada jaringan juga bertambah besar. Sehingga dibuatlah sebuah permodelan adaptif relai yang dapat mengatasi masalah proteksi jaringan tersebut.

Untuk dapat mengubah *setting* otomatis sesuai sumber diperlukan alat yang dapat memberitahu relai bahwa kondisi sumber telah berubah. Dengan menggunakan komunikasi *master slave*, maka informasi kondisi sumber dapat dikirimkan ke relai yang lain. Perubahan *setting* ini akan diatur oleh program yang telah dimasukkan ke *microcontroller*. Permodelan ini juga dilengkapi dengan sistem monitoring adaptif relai agar dapat mempermudah pengguna dalam memeriksa kerja adaptif relai. Pada sistem

monitoring adaptif relai terdapat informasi seperti bentuk grafik dan indikator kondisi arus yang telah terjadi pada sistem.

## 1.2 Permasalahan

Adapun permasalahan yang akan diangkat sebagai bahan Tugas Akhir ini adalah bahwa penambahan beban yang ada harus diiringi dengan penambahan sumber, agar beban yang ada dapat teraliri listrik dengan maksimal. Dengan adanya penambahan sumber, arus yang ada pada saluran akan bertambah besar. Sehingga dibutuhkan relai yang dapat mengubah *setting* waktunya sendiri berdasarkan kondisi sumber saat itu. Maka dari itu, pada tugas akhir ini akan membuat permodelan komunikasi adaptif relai beserta sistem monitornya dengan menggunakan Arduino Uno sebagai pemrosesan komunikasi antar relai.

## 1.3 Batasan Masalah

Agar penulisan buku Tugas Akhir ini tidak menyimpang dan mengambang dari tujuan yang semula direncanakan sehingga mempermudah mendapatkan data dan informasi yang diperlukan, maka penulis menetapkan batasan-batasan masalah sebagai berikut :

- a. Deteksi jumlah sumber
- b. Komunikasi *master slave* adaptif relai
- c. Sistem monitor adaptif relai

## 1.4 Tujuan

Pembuatan permodelan komunikasi dan monitoring adaptif relai bertujuan untuk membuat permodelan perancangan komunikasi adaptif relai arus lebih, yang dilengkapi pemantuan pada *Personal Computer* yang diharapkan mampu mempermudah para operator dalam mengetahui kondisi *real* tiap relai.

## 1.5 Metodologi Penelitian

Pembuatan Tugas Akhir ini dilakukan melalui beberapa tahapan metodologi, yaitu studi literatur, perancangan sistem, pengambilan data percobaan, analisis data dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur Tugas Akhir Sistem Komunikasi Dan Monitoring Adaptif Relai Arus Lebih Terhadap Sumber Tegangan

Pada Saluran Listrik Satu Fasa ini akan mempelajari sistematika proteksi pada tegangan listrik 1 fasa, mempelajari prinsip kerja OCR atau *Over Current Relay* dan juga mempelajari tentang pengiriman data *master slave* untuk konfigurasi jumlah sumber dan *software* yang digunakan untuk monitor relai.

Pada tahap perancangan sistem terdiri dari tiga yaitu, perancangan mekanik, perancangan sistem elektrik dan perancangan software. Pada tahap perancangan mekanik terdiri dari perancangan box tempat sistem elektrik sebagai relai indikator. Sedangkan perancangan sistem elektrik terdiri dari Arduino UNO, relai AC 220 Volt dan modul komunikasi RS485. Pada perancangan software yaitu pembuatan program komunikasi data pada Arduino UNO ke Arduino Mega 2560 dan pembuatan software monitor relai pada LabVIEW. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

## **1.6 Sistematika Laporan**

Sistematika pembahasan Tugas Akhir ini terdiri dari lima bab, yaitu pendahuluan, teori penunjang, perencanaan dan pembuatan alat, pengujian dan analisa alat, serta penutup.

### **BAB I : PENDAHULUAN**

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

### **BAB II : TEORI PENUNJANG**

Berisi teori penunjang yang mendukung dalam perencanaan dan pembuatan alat.

### **BAB III : PERANCANGAN ALAT**

Membahas tentang perencanaan dan pembuatan perangkat keras yang meliputi rangkaian-rangkaian, desain bangun, dan perangkat lunak yang meliputi program yang akan digunakan untuk mengaktifkan alat tersebut.

### **BAB IV : PENGUJIAN DAN ANALISA ALAT**

Membahas tentang pengukuran, pengujian, dan penganalisaan terhadap kepresisian sensor dan alat yang telah dibuat.

## **BAB V : PENUTUP**

Menjelaskan tentang kesimpulan dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

### **1.7 Relevansi**

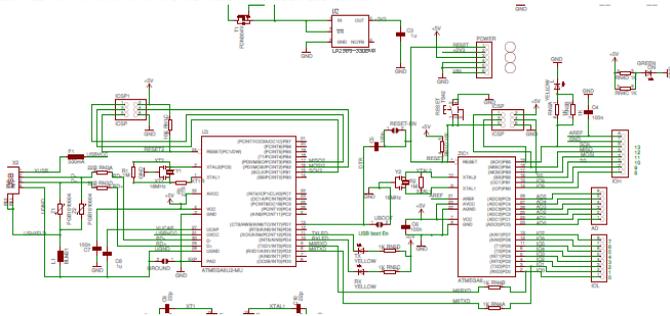
Diharapkan alat ini dapat terealisasi, alat ini dapat digunakan untuk untuk mempermudah proses pembelajaran dalam koordinasi komunikasi dan monitoring terhadap relai arus lebih (*Over Current Relay*).

## BAB II TEORI DASAR

Pada Bab II ini akan dijelaskan mengenai teori-teori dasar yang menunjang dan berhubungan dalam pengerjaan Tugas Akhir ini. Teori dasar ini diharapkan mampu membantu dalam pengerjaan Tugas Akhir dan dapat dijadikan referensi nantinya.

### 2.1 Arduino UNO[2]

Arduino UNO adalah papan mikrokontroler berdasarkan ATmega328. Arduino ini memiliki 14 pin *input / output* digital (yang 6 dapat digunakan sebagai output PWM), 6 *input* analog, resonator keramik 16 MHz, koneksi USB, colokan listrik, header ICSP, dan tombol reset. Ini berisi semua yang dibutuhkan untuk mendukung mikrokontroler. Cukup hubungkan ke komputer dengan kabel USB atau nyalakan dengan adaptor AC-ke-DC atau baterai untuk memulai.



**Gambar 2.1** Board Arduino UNO

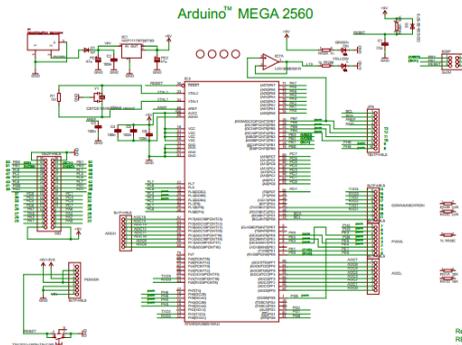
Spesifikasi Arduino Mega 2560 adalah sebagai berikut:

1. *Microcontroller* ATmega328
2. Tegangan operasi 5V
3. Tegangan input (yang direkomendasikan) 7-12V
4. Tegangan input (batas) 6-20V
5. Pin digital I/O 14 (6 dapat digunakan sebagai output PWM)
6. 6 pin analog *input*
7. Arus DC pin I/O Pin 40 mA
8. Arus DC untuk 3.3V Pin 50 mA

9. *Flash Memory* 32 KB (ATmega328) yang 0.5 KB digunakan sebagai *bootloader*
10. SRAM 2 KB (ATmega328)
11. EEPROM 1 KB (ATmega328)
12. *Clock Speed* 16 MHz

## 2.2 Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega2560. Arduino ini memiliki 54 digital pin input / output (yang 15 dapat digunakan sebagai output PWM), 16 analog input, 4 UART (hardware port serial), 16 MHz osilator kristal, koneksi USB, jack listrik, header ICSP, dan tombol reset. Arduino ini berisi semua yang diperlukan untuk mendukung mikrokontroler; untuk dapat terhubung ke komputer dengan menggunakan kabel USB atau sumber tegangan berasal dari adaptor AC-DC atau baterai untuk menghidupkan arduino .



**Gambar 2.2** Board Arduino Mega 2560

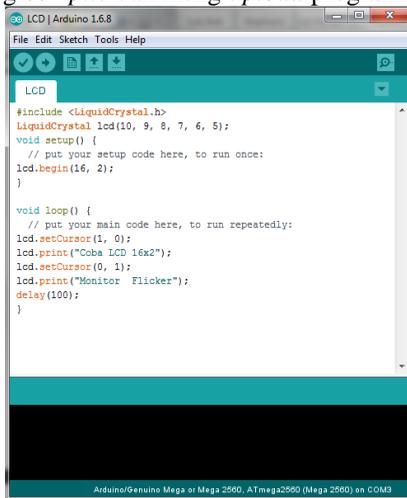
Spesifikasi Arduino Mega 2560 adalah sebagai berikut:

1. Menggunakan chip microcontroller AtMega2560.
2. Tegangan operasi 5 Volt.
3. Tegangan input (yang direkomendasikan, via jack DC) sebesar 7-12 Volt.
4. Digital I/O sebanyak 54 buah, 6 diantaranya menyediakan PWM output.
5. Analog input pin sebanyak 16 buah.
6. Arus DC per pin I/O sebesar 20 mA.

7. Arus DC pada pin 3,3 Volt sebesar 50 mA.
8. Flash memory sebesar 256 KB, 8 KB telah digunakan untuk bootloader.
9. SRAM sebesar 8 kb.
10. EEPROM sebesar 4 kb.
11. Clock speed sebesar 16 Mhz.
12. Dimensi Arduino Mega 2560 sebesar 101,5 mm x 53,4 mm.
13. Berat Arduino Mega 2560 sebesar 37 g.

### 2.3 Arduino IDE[3]

*Board* Arduino dapat diprogram menggunakan *software open source* bawaan Arduino IDE. Arduino IDE adalah sebuah aplikasi *crossplatform* yang berbasis bahasa pemrograman *processing* dan *wiring*. Arduino IDE didesain untuk mempermudah pemrograman dengan adanya kode editor yang dilengkapi dengan *syntax highlighting*, *brace matching*, dan indentasi otomatis untuk kemudahan pembacaan program, serta dapat *meng-compile* dan *meng-upload* program ke *board* dalam satu klik.



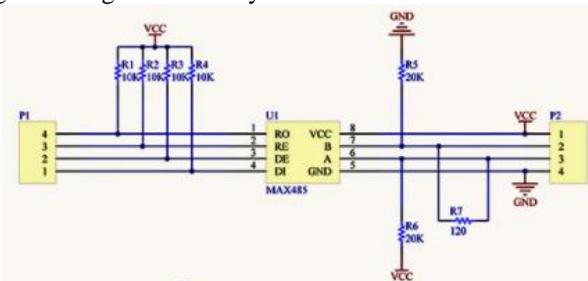
**Gambar 2.3** Jendela Arduino IDE

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. *Editor* program, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah *microcontroller* tidak akan bisa memahami bahasa *processing*. Yang bisa dipahami oleh *microcontroller* adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan Arduino.

## 2.4 Modul Komunikasi RS485[4]

RS485 adalah teknik komunikasi data serial yang dikembangkan di tahun 1983 dimana dengan teknik ini, komunikasi data dapat dilakukan pada jarak yang cukup jauh yaitu 1,2 Km dengan kecepatan dapat mencapai 20 Mbps. Berbeda dengan komunikasi serial RS232 yang mampu berhubungan secara *one to one*, maka komunikasi RS485 selain dapat digunakan untuk komunikasi *multidrop* yaitu berhubungan secara *one to many* dengan jarak yang jauh teknik ini juga dapat digunakan untuk menghubungkan 32 unit beban sekaligus hanya dengan menggunakan dua buah kabel saja tanpa memerlukan referensi *ground* yang sama antara unit yang satu dengan unit lainnya.



**Gambar 2.4** Modul Komunikasi RS485

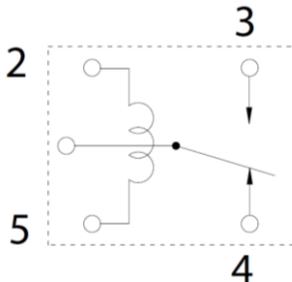
Pin yang ada pada RS485 adalah sebagai berikut :

- DI (*Data In*) Data pin DI ditransmisikan pada baris A & B saat modul berada dalam mode *Transmit* (mengirim data). Untuk mengatur modul dalam mode pengiriman nilai DE dibuat 1 dan RE dibuat 1. Pin DI terhubung ke pin Tx *Host Microcontroller* UART.
- RE (*Receive Enable*) RE pin digunakan untuk mengkonfigurasi modul dalam *Receive Mode* (menerima data).

- DE (*Data Enable*) DE pin digunakan untuk mengkonfigurasi modul dalam *Mode Transmitt*. Untuk mengfungsikan RS485 dalam mode *Transmit* dan *Receive* pin DE dan RE dihubungkan menjadi 1.
- RO (*Receive Out*) data yang diterima pada pin A & B diberikan pada pin RO. Pin RO terhubung ke pin Rx dari mikrokontroler.
- A & B (*Differential Input and Ouput Pins*) data ditransmisikan dan diterima pada garis A & B.

## 2.5 Relay AZ943[5]

*Relay* adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen elektromagnetik yang terdiri dari 2 bagian utama yakni elektromagnet (*coil*) dan mekanikal (seperangkat kontak saklar/*switch*). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.

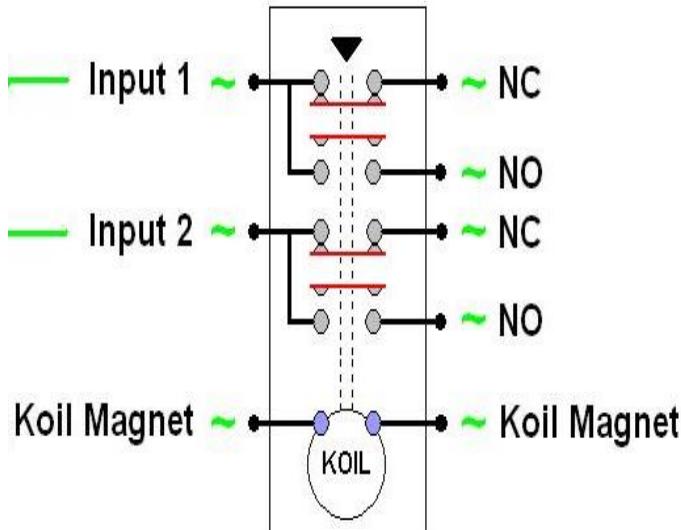


**Gambar 2.5** Relay AZ943

AZ943 merupakan *relay* yang bekerja berdasarkan prinsip elektromagnetik, dengan tegangan operasi pada *coil* sebesar 5VDC dan arus sebesar 71,4 mA. Dengan tegangan 5 VDC, dapat membangkitkan tegangan pada kontak sebesar 30VDC dan 277VAC. waktu operasi pada relay ini adalah 10ms (maks) dan waktu rilisnya 5ms(maks).

## 2.6 Relay AC MY2 220 Volt[6]

Fungsi dan cara kerja dari relai AC ini hampir sama dengan relai DC. Yang membedakan adalah sumber yang digunakan untuk mengaktifkan relai. Relai AC membutuhkan arus bolak balik, dengan tegangan antara 6 – 240 VAC.



**Gambar 2.6** Relay AC MY2 220 Volt

Relay AC MY2 memiliki spesifikasi sebagai berikut:

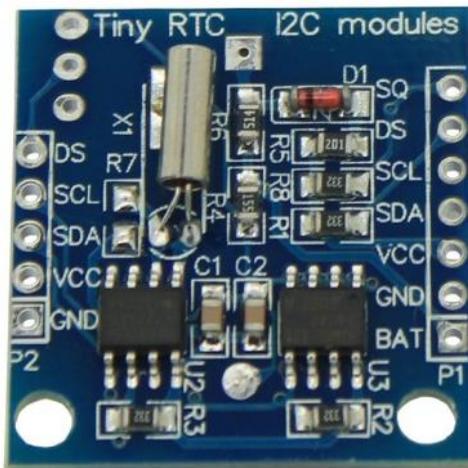
1. Tegangan kerja 220/240 VAC
2. Arus Kerja pada 50 Hz adalah 4.8/5.3 mA dan pada 60 Hz adalah 4.2/4.6 mA
3. Memiliki resistansi kumparan sebesar 18,790  $\Omega$
4. Memiliki 8 kaki. 2 untuk *coil*, 2 untuk kontak dan memiliki 2 *Normally Open* dan 2 *Normally Close*

### 2.7 Real Time Clock DS1307 modules (RTC)[7]

*Real Time Clock* (RTC) adalah jenis pewaktu yang bekerja berdasarkan waktu yang sebenarnya atau dengan kata lain berdasarkan waktu yang ada pada jam kita. Meskipun istilah sering mengacu pada perangkat di komputer pribadi, server dan *embedded system*, RTC hadir di hampir semua perangkat elektronik yang perlu untuk menjaga keakuratan waktu. RTC memiliki sumber tenaga alternatif, sehingga mereka dapat terus menjaga waktu sementara sumber utama daya mati atau tidak tersedia. Sumber tenaga alternatif ini biasanya berupa baterai lithium dalam sistem lama, tetapi beberapa sistem yang lebih baru menggunakan supercapacitor, karena mereka dapat diisi ulang dan dapat

disorder. Sumber daya alternatif juga dapat menyalurkan listrik ke RAM yang didukung baterai. Pada umumnya tenaga alternatif yang digunakan sebesar 3 Volt dari baterai lithium.

Kebanyakan RTC menggunakan osilator kristal, tetapi beberapa menggunakan frekuensi saluran listrik. Dalam banyak kasus frekuensi osilator yang digunakan adalah 32,768 kHz. Frekuensi ini sama dengan yang digunakan dalam jam kuarsa dan jam tangan, selain itu frekuensi yang dihasilkan adalah persis 215 siklus per detik, yang merupakan tingkat nyaman untuk digunakan dengan sirkuit biner sederhana.



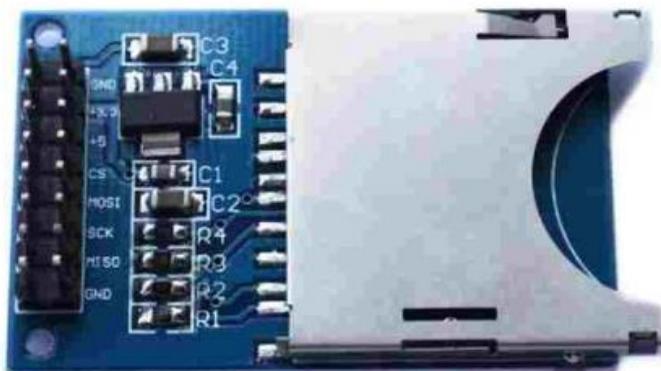
**Gambar 2.7** RTC Tiny I2C modules

Modul RTC kecil ini didasarkan pada chip jam DS1307 yang mendukung protokol I2C. RTC ini menggunakan sel baterai Lithium (CR1225). Jam / kalender menyediakan detik, menit, jam, hari, tanggal, bulan, dan Informasi tahun. Akhir tanggal bulan secara otomatis disesuaikan dengan bulan dengan kurang dari 31 hari, termasuk koreksi untuk tahun kabisat. Jam beroperasi baik dalam 24 jam atau 12 jam dengan format AM / PM indikator.

## **2.8 Modul SD Card / MMC[7]**

Merupakan suatu modul untuk mempermudah antarmuka antara SD Card (atau MMC) dan mikrokontroler dengan tegangan kerja +5 VDC. SD Card (atau MMC) dapat digunakan sebagai memori yang dapat

diganti dengan mudah sehingga memudahkan dalam ekspansi ke kapasitas memori yang lebih besar. Tersedia Ferroelectric Nonvolatile RAM (FRAM) yang dapat digunakan sebagai *buffer* sementara dalam mengakses SD Card (atau MMC) atau sebagai tempat penyimpanan data lain. Modul ini dapat digunakan antara lain sebagai penyimpan data pada sistem absensi, sistem antrian, atau aplikasi *datalogging* lainnya.



**Gambar 2.8** Modul SD Card / MMC

Modul SD Card / MMC ini memiliki spesifikasi kerja *hardware* sebagai berikut :

1. Tegangan *supply* +5 VDC.
2. Jenis kartu yang didukung: SD Card (dan MMC).
3. Antarmuka SD Card (dan MMC) dengan mikrokontroler secara SPI.
4. Tersedia 2 KByte Ferroelectric Nonvolatile RAM FM24C16.
5. Antarmuka FRAM dengan mikrokontroler secara *TwoWire Interface*.
6. Tersedia contoh aplikasi untuk DT-51™ *Low Cost Series* dan DT-AVR *Low Cost Series* dalam bahasa *BASIC* untuk MCS-51® (BASCOM-8051©) dan bahasa C untuk AVR® (CodeVisionAVR©).
7. Kompatibel dengan DT-51™ *Low Cost Series* dan DT-AVR *Low Cost Series*. Mendukung DT-51™ *Minimum System* (MinSys) ver 3.0, DT-51™ *PetraFuz*, dan lain-lain.

## 2.9 *Miniature Circuit Breaker (MCB)*[7]

*Miniature Circuit Breaker (MCB)* memiliki fungsi sebagai alat pengaman arus lebih. MCB ini memproteksi arus lebih yang disebabkan terjadinya beban lebih dan arus lebih karena adanya hubungan pendek. Prinsip dasar kerjanya yaitu untuk pemutusan hubungan yang disebabkan beban lebih dengan relai arus lebih sesaat menggunakan elektromagnet.

Bila elektromagnet bekerja, maka akan memutus hubungan kontak yang terletak pada pemadam busur dan membuka saklar. MCB untuk rumah seperti pada pengaman lebur diutamakan untuk proteksi hubungan pendek, sehingga pemakaiannya lebih diutamakan untuk mengamankan instalasi atau konduktornya. Arus nominal yang digunakan pada APP dengan mengenal tegangan 230/400 V ialah: 1 A, 2 A, 4 A, 6 A, 10 A, 16 A, 20 A, 25 A, 35 A, dan 50 A disesuaikan dengan tingkat VA konsumen. Adapun kemampuan membuka (*breaking capacity*) bila terjadi hubung singkat 3 KA dan 6 KA (SPLN 108-1993). MCB yang khusus digunakan oleh PLN mempunyai tombol biru. MCB pada saat sekarang ini paling banyak digunakan untuk instalasi rumah, instalasi industri maupun instalasi gedung bertingkat.

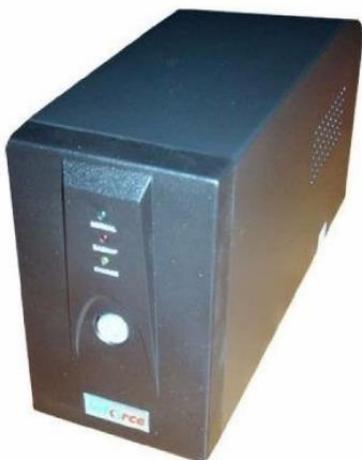


**Gambar 2.9** *Miniature Circuit Breaker (MCB)*

## 2.10 *Uninterruptible Power Supply (UPS) Inforce 650 VA*[8]

UPS adalah singkatan dari *uninterruptible power supply* sebagai alat *back-up* listrik ketika PC atau kehilangan energi dari sumber utamanya. UPS bekerja diantara *device* yang di suplai dan colokan listrik, dari colokan listrik yang di alirkan ke baterai yang berada pada UPS dan kemudian di simpan untuk kestabilan tegangan energi. Listrik yang di

simpan pada baterai akan di pakai ketika sumber energi utama listrik terputus.



**Gambar 2.10** *Uninterruptible Power Supply (UPS) Inforce 650 VA*

*Uninterruptible Power Supply (UPS) Inforce 650 VA* memiliki spesifikasi sebagai berikut :

1. Kapasitas daya 650 VA
2. *Range* input 140V – 300VAC
3. Stabilitas *output* 230 V + /- 10%
4. 3 Step AVR
5. Efisiensi tinggi pada mode *in-line* 95%
6. Tipe baterai 8.2 Ah
7. 2 soket output
8. Indikator LCD

### **2.11 Liquid Crystal Display Keypad (LCD Keypad)[9]**

LCD (*Liquid Crystal Display*) berfungsi menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi microcontroller. Sumber cahaya di dalam sebuah perangkat LCD adalah lampu neon berwarna putih dibagian belakang susunan kristal cair tadi.

Titik cahaya yang jumlahnya puluhan ribu bahkan jutaan inilah yang membentuk tampilan. Kutub kristal cair yang dilewati arus listrik akan

berubah karena pengaruh polarisasi medan magnetik yang timbul dan oleh karenanya akan hanya beberapa warna.

LCD membutuhkan driver supaya bisa dikoneksikan dengan sistem minimum dalam suatu microcontroller. Driver yang disebutkan berisi rangkaian pengaman, pengatur tingkat kecerahan maupun data, serta untuk mempermudah pemasangan di microcontroller.



**Gambar 2.11** *Liquid Crystal Diplay Keypad (LCD Keypad)*

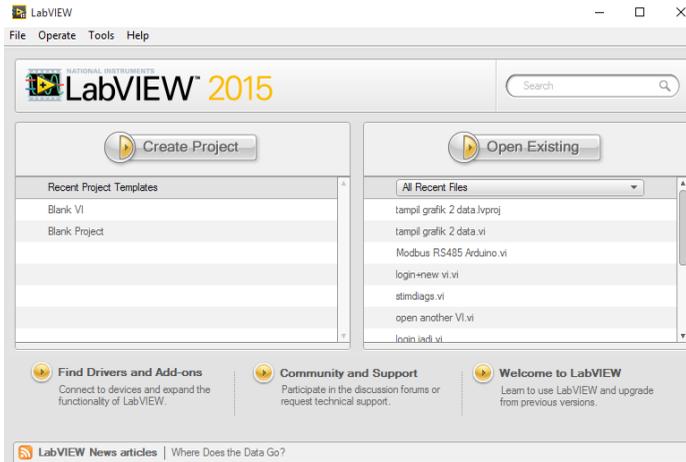
LCD Keypad dikembangkan untuk Arduino *Shield*, tujuannya untuk menyediakan antarmuka yang *user-friendly* dan memungkinkan pengguna untuk membuat berbagai pilihan menu dan lainnya. LCD Keypad ini terdiri dari 1602 karakter putih *backlight* biru. Terdapat 5 tombol yang terdiri dari *select*, *up*, *right*, *down* dan *left*. Untuk menyimpan pin IO digital, antarmuka keypad hanya menggunakan satu saluran ADC. Berikut adalah konfigurasi pin untuk LCD Keypad.

**Tabel 2.1** Konfigurasi Pin LCD Keypad 16x2

Pin	Function
Analog 0	Button (select, up, right, down and left)
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (Data or Signal Display Selection)
Digital 9	Enable
Digital 10	Backlit Control

## 2.12 Software LabVIEW[10]

LabVIEW adalah sebuah *software* pemrograman yang diproduksi oleh National Instruments dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau Visual Basic, LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa labVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis text.



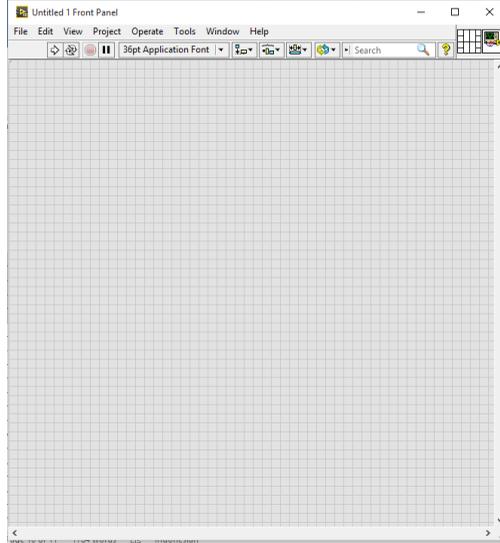
**Gambar 2.12** Tampilan Awal LabVIEW

Program LabVIEW dikenal dengan sebutan VI atau Virtual Instruments karena penampilannya dan operasinya dapat menyerupai sebuah instrumen. Pada LabVIEW, *user* pertama-tama diharuskan membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push buttons*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LED dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu *user* menyusun blok diagram yang berisi kode-kode VI untuk mengontrol front panel. Software LabVIEW terdiri dari tiga komponen utama, yaitu :

1. *Front Panel*

*Front panel* adalah bagian window yang berlatar belakang abu-abu serta mengandung kontrol dan indikator. *front panel* digunakan

untuk membangun sebuah VI, menjalankan program dan mendebug program. Tampilan dari front panel dapat di lihat pada Gambar 2.8

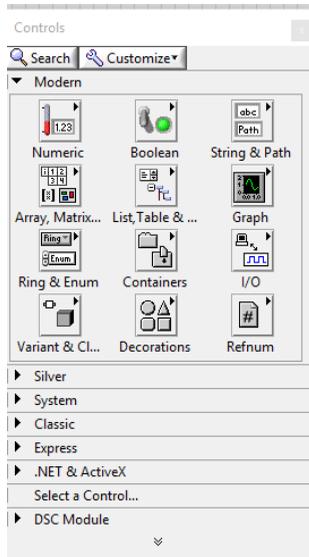


**Gambar 2.13** Front Panel

## 2. Blok diagram dari Vi

Blok diagram adalah bagian *window* yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai instruksi untuk *front panel*. Tampilan dari blok diagram dapat lihat pada Gambar 2.14

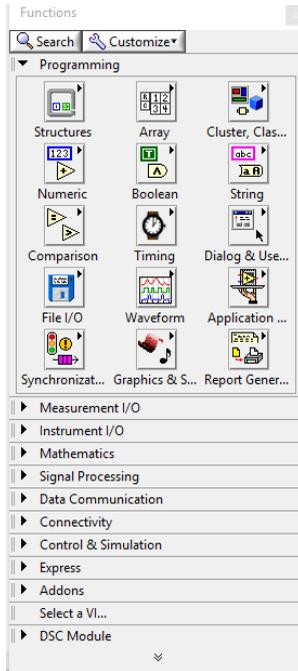




**Gambar 2.15** Control Palette

b. *Functions Palette*

*Functions Palette* di gunakan untuk membangun sebuah blok diagram, *functions palette* hanya tersedia pada blok diagram, untuk menampilkannya dapat dilakukan dengan mengklik *windows >> show control palette* atau klik kanan pada lembar kerja blok diagram. Contoh dari *functions palette* ditunjukkan pada Gambar 2.16



**Gambar 2.16** *Functions palette*

## **BAB III**

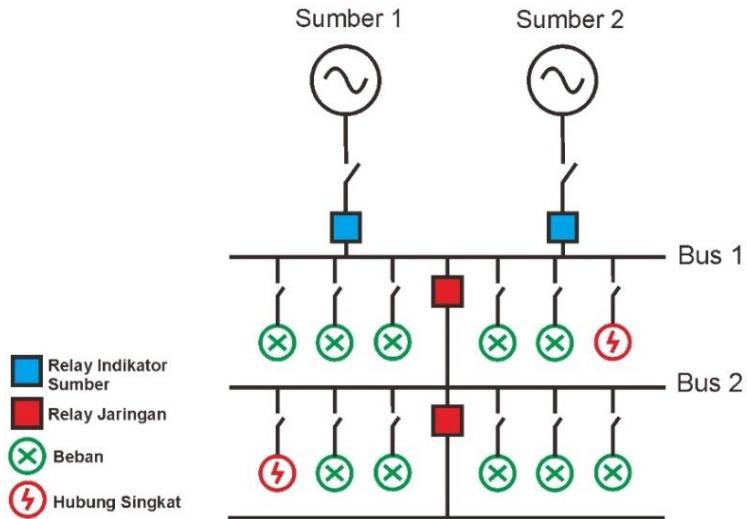
### **PERENCANAAN ALAT**

Pada bab ini akan dibahas mengenai perancangan serta pembuatan “Sistem Komunikasi dan Monitoring Adaptif Relai Arus Lebih Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa”, baik perancangan perangkat elektronik (*hardware*), perancangan dan pembuatan perangkat lunak (*software*) yang meliputi :

1. Perancangan Mekanik Relai Indikator
2. Perancangan *Hardware* terdiri dari :
  - a Perancangan Relai Indikator
  - b Perancangan Alat Komunikasi RS485
  - c Perancangan LCD *Keypad*
  - d Perancangan Rangkaian RTC
  - e Perancangan Rangkaian SD Card
3. Perancangan *Software* yang berupa *flowchart* terdiri dari :
  - a Pemrograman Arduino IDE
  - b Pemrograman *Interface* dengan LabVIEW

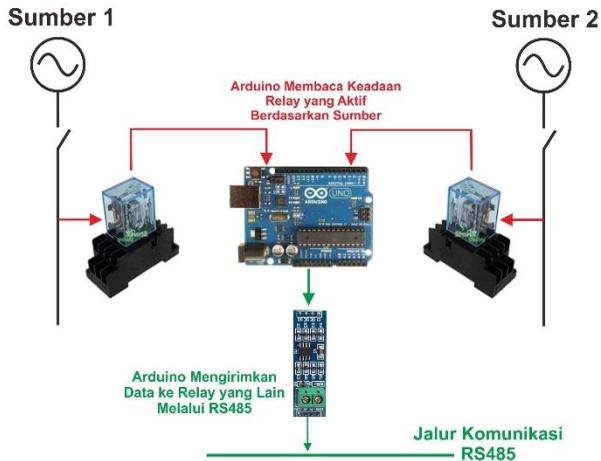
#### **3.1 Diagram Fungsional Alat**

Perencanaan Tugas Akhir “Sistem Komunikasi dan Monitoring Adaptif Relai Arus Lebih Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa” ini mengenai sistem kerja alat secara keseluruhan. Dimana pada alat ini berfungsi sebagai pendeteksi jumlah sumber tegangan yang nantinya akan mempengaruhi sistem koordinasi adaptif relai dan untuk monitor kondisi relai. Komunikasi pada Tugas Akhir ini adalah komunikasi antar relai indikator dan relai utama yang bertugas untuk memproteksi jaringan. Komunikasi diperlukan agar relai utama dapat mengetahui jumlah sumber yang ada pada jaringan. Dan monitoring merupakan *interface* yang menghubungkan antara sistem dengan manusia atau *user*. *Interface* ini akan menampilkan kondisi yang terjadi pada relai utama secara *realtime*. Sedangkan adaptif berarti bahwa *setting* waktu relai ini dapat berubah kondisi sesuai dengan keadaan sumber dan arus, jika beban dan sumber pada sistem bertambah maka arus dalam jaringan juga bertambah, agar sistem tetap berjalan dan beban tetap dapat beroperasi maka *setting* dari relai harus berubah. Berikut merupakan skema sistem secara keseluruhan.



**Gambar 3.1** Skema Sistem Secara Keseluruhan

Untuk dapat mengetahui apakah sumber yang digunakan hanya 1 sumber atau 2 sumber, dideteksi dengan menggunakan relai indikator. Relai indikator ini berfungsi untuk memeriksa apakah sumber yang tersambung pada jaringan hanya 1 sumber atau 2 sumber. Dengan memasang sebuah relai AC pada masing-masing saluran sumber, dapat diketahui berapa jumlah sumber yang aktif. Relai AC ini akan aktif ketika jaringan yang dipasangi relai teraliri arus. Ketika relai AC berubah kondisi, arduino akan mengetahui berapa jumlah sumber saat ini yang aktif. Dan selanjutnya akan mengirimkan data sumber ke relai utama agar dapat menentukan berapa besar *setting* arus yang harus ditetapkan untuk proteksi jaringan. Gambar 3.2 berikut merupakan skema relai indikator yang digunakan pada Tugas Akhir ini.

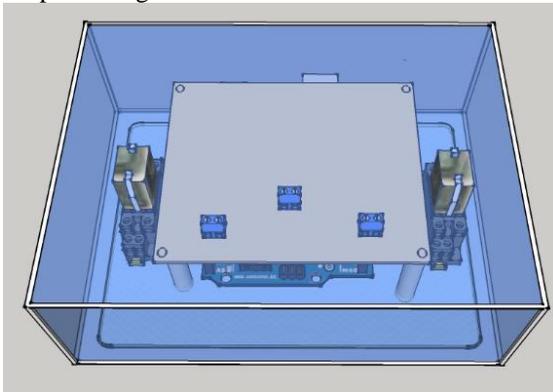


**Gambar 3.2** Skema Relai Indikator

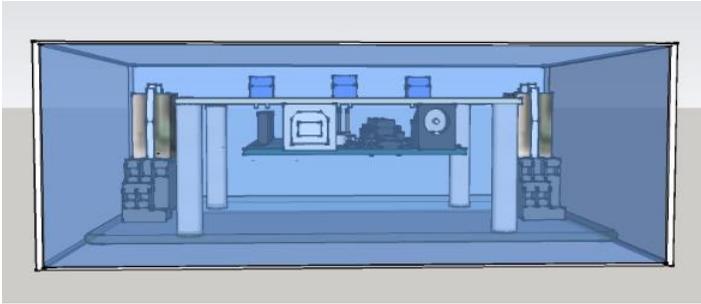
### 3.2 Perancangan Mekanik

Perancangan mekanik pada tugas akhir ini berupa perancangan pada bagian relai indikator.

Pada perancangan mekanik relai indikator merupakan kotak *acrylic* yang didalamnya terdapat rangkaian Relai 220 Volt AC, *Pull Down Resistor*, *Shield Arduino Uno* dan Komunikasi RS485. Berikut adalah gambar perancangan mekanik untuk relai indikator.



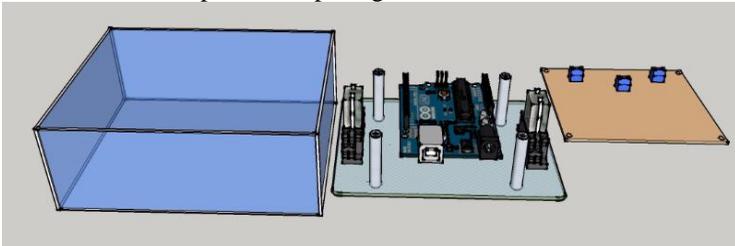
**Gambar 3.3** Rancangan Mekanik Relai Indikator Tampak Atas



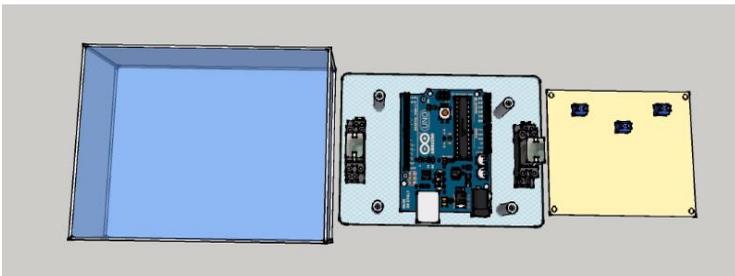
**Gambar 3.4** Rancangan Mekanik Relai Indikator Tampak Samping

### 3.3 Perancangan Elektronik

Pada perancangan elektronik (*hardware*) pada tugas akhir ini merupakan perancangan komponen yang digunakan pada relai indikator. Untuk perancangan elektronik (*hardware*) relai indikator komponen penyusunnya terdapat rangkaian Relai, *Pull Down Resistor*, *Shield* Arduino Uno dan Komunikasi RS485. Tampilan perancangan *hardware* untuk relai indikator dapat dilihat pada gambar dibawah ini :



**Gambar 3.5** Rancangan Elektronik Relai Indikator Tampak Samping



**Gambar 3.6** Rancangan Elektronik Relai Indikator Tampak Atas

Susunan dari perancangan elektronik terbagi menjadi 2 bagian, yaitu, bagian atas dan bagian bawah. Pada bagian bawah terdapat *shield* Arduino Uno sebagai kontrol relai indikator dan 2 buah relai AC. Sedangkan pada bagian atas terdapat modul komunikasi RS485 yang berfungsi untuk mengirimkan data jumlah sumber ke relai utama.

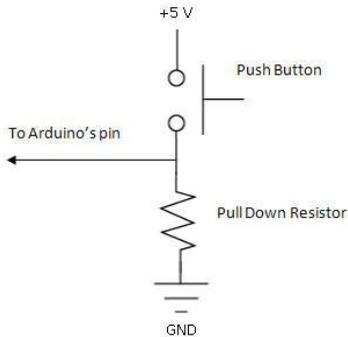
Tata letak dari komponen-komponen sistem kontrol ini dibuat seperti demikian untuk mempermudah dalam pengoperasiannya. Selain itu dengan diletakkan menjadi satu *board* seperti Gambar 3.5 dan Gambar 3.6 diatas akan menjadi lebih rapi dan efisien. Mengenai komponen – komponen yang digunakan akan dijelaskan pada sub bab berikut ini.

### **3.3.1 Perancangan Relai Indikator**

Adaptif relai merupakan relai yang *setting*-nya dapat berubah sesuai dengan penambahan sumber dan penambahan arus yang ada pada jaringan. Agar dapat menghasilkan relai yang bersifat adaptif, maka pada sisi sumber harus ditambahkan alat yang berguna untuk mendeteksi ada atau tidaknya sumber dan berapa jumlah sumber yang mensuplai jaringan. Alat ini sangat penting karena agar adaptif relai dapat berjalan, jumlah sumber yang ada sangatlah berpengaruh pada besar arus *setting* yang ada pada relai utama.

Ketika beban bertambah, arus yang lewat pada jaringan menjadi lebih besar. Jika arus yang lewat lebih besar daripada arus *setting* maka relai utama akan *trip*. Agar hal ini tidak terjadi, maka perlu ditambah sumber, agar beban yang ada dapat tersuplai semua. Maka jika sumber ditambah, arus *setting* pada relai utama juga harus bertambah.

Pada tugas akhir ini, untuk dapat mewujudkan alat yang dapat mengetahui ada tidaknya sumber maka dirancanglah sebuah relai indikator yang dapat mewujudkan fungsi tersebut. Relai AC 220 V dipasang pada sisi saklar sumber, maka ketika sumber *on* atau *off* relai AC ini akan dapat membaca kondisi sumber. Pada kedua sumber dipasangkan relai AC yang nantinya terhubung dengan pin digital 4 dan digital 7 Arduino. Dan sebagai indikator aktifnya sumber menggunakan LED yang terhubung pada pin digital 5 dan digital 6 Berikut adalah rangkaian relai indikator dengan Arduino :



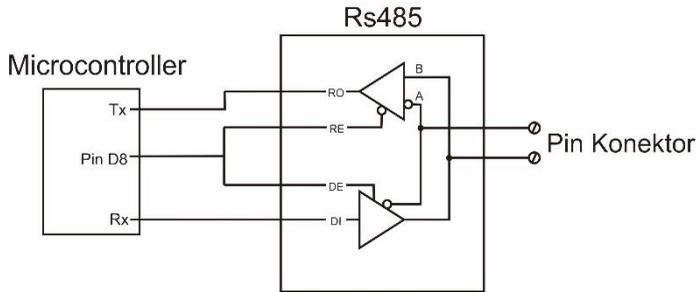
**Gambar 3.7** Rangkaian Resistor *Pull Down* dengan Arduino

Agar arduino dapat mendeteksi perubahan yang ada pada relai maka dipasang resistor *pull down*. Sehingga jika relai dalam kondisi *Normally Open* atau pada gambar dimisalkan oleh *push button*, VCC 5 Volt tidak akan mengalirkan tegangan dan pin Arduino terhubung dengan *Ground* yang akan menghasilkan *logic 0* pada Arduino. Sedangkan ketika relai atau *push button* dalam kondisi *Normally Close*, VCC 5 Volt akan mengalir ke pin arduino (tanpa hambatan/resistor) sehingga pin Arduino akan membaca 5 Volt atau *logic 1*. Resistor yang dipasang pada *ground* ini berfungsi untuk membatasi arus *short circuit* agar tidak terlalu besar dan dapat menimbulkan kerusakan pada Arduino.

### 3.3.2 Perancangan Komunikasi RS485

Dalam perancangan tugas akhir ini diperlukan alat komunikasi antara relai dengan relai dan antara relai dengan komputer. Komunikasi yang digunakan harus dapat mengirimkan data dengan jarak jauh, karena dalam kondisi real relai akan dipasang tidak berdekatan.

Protokol Komunikasi Serial RS485 mampu mentransmisikan data hingga jarak maksimum  $\pm 4000\text{m}$  atau 4km. Data ditransmisikan oleh 2 buah kabel, yaitu A dan B. Untuk dapat mencapai transmisi data sepanjang 4 km, media pembawa data harus lancar dan panjang garis A dan B harus jarak jauh. Berikut adalah gambar sambungan antara Arduino dengan RS485.



**Gambar 3.8** Schematic RS485 dengan Arduino Mega dan Arduino UNO

Untuk menyambungkan RS485 dengan arduino menggunakan pin Rx, Tx dan pin digital. Pada Arduino Mega, pin DI (*data in*) RS485 terhubung ke pin 18 Arduino sebagai Tx1 *software serial*. Pin RO (*receive out*) ke pin 19 Arduino sebagai Rx1 *software serial*. Pin DE (*data enable*) dan RE (*receive enable*) di jumper dan dihubungkan ke pin 8 Arduino. Pin A dan B (RS485 *pair*) dipasang kedua modul RS485. Dan untuk Vcc dan GND RS485 dihubungkan pada Vcc dan GND Arduino. Berikut merupakan *schematic* sambungan RS 485 dengan Arduino Mega dan Arduino UNO.

### 3.3.3 Perancangan LCD Keypad

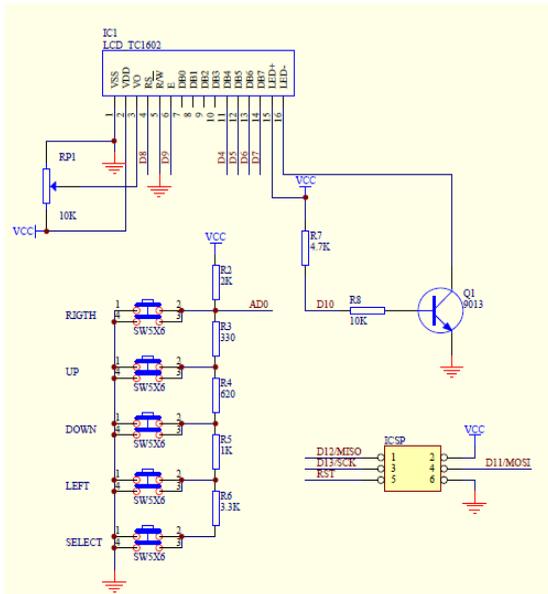
Pada Tugas Akhir ini *Liquid Crystal Display* atau LCD digunakan untuk mempermudah *user* untuk melihat kondisi terkini dari relai. LCD yang digunakan adalah LCD Keypad 16 x 2. Menu yang akan ditampilkan ada 3, yaitu :

1. Menu *Display* yang berisi I *setting*, I rms dan V rms
2. Menu Status yang berisi Waktu RTC dan Status Jaringan
3. Menu *Short Circuit* yang berisi I *setting*, I rms, Karakteristik Relai, Tes *Trip*, Reset Relai dan Menu Awal

Berikut adalah pengkabelan yang harus dilakukan untuk menghubungkan LCD Keypad dengan Arduino Mega 2560 :

1. Pin A0 dengan pin 5 tombol LCD Keypad
2. Pin Digital 4 dihubungkan dengan pin DB 4
3. Pin Digital 5 dihubungkan dengan DB5
4. Pin Digital 6 dihubungkan dengan DB 6
5. Pin Digital 7 dihubungkan dengan DB7
6. Pin Digital 8 dihubungkan dengan pin RS

7. Pin Digital 9 dihubungkan dengan pin *Enable*
8. Pin Digital 10 dihubungkan dengan pin *Backlight Control*



**Gambar 3.9** Schematic LCD Keypad dengan Arduino

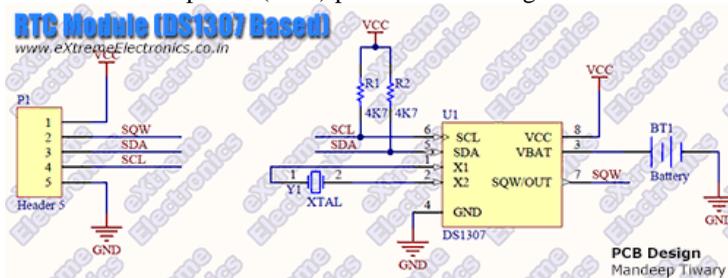
### 3.3.4 Perancangan RTC DS1307 dan *Data Logger*

Pada Tugas Akhir ini membutuhkan pencatatan data arus dan tegnagn serta kondisi yang tengah terjadi pada jaringan secara *real time*. Ini berfungsi jika nantinya *user* ingin melihat histori yang telah terjadi pada relai. Untuk dapat mencatat kondisi secara real time dibutuhkan penghitungan waktu yang akurat. Penghitung ini harus dapat menghitung jam, menit, detik, tanggal, bulan dan tahun secara akurat. Untuk penghitungan waktu tersebut digunakan modul RTC DS1307. Dimana modul ini sudah memiliki baterai sebagai sumber ketika sumber utamanya habis atau tidak ada. Sehingga meski sumber terputus, penghitung waktu akan tetap berjalan.

Modul RTC DS1307 dapat langsung dihubungkan pada arduino. Sambungan dengan Arduino Mega dapat dituliskan sebagai berikut :

1. VCC – pin 5V Arduino Mega
2. GND – pin *Ground* Arduino Mega

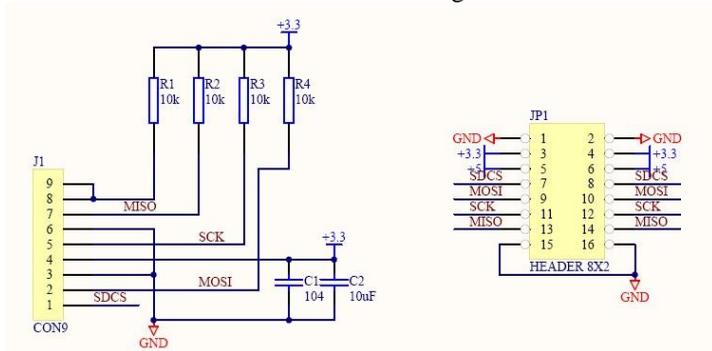
3. SDA – pin 20 (SDA) pada Arduino Mega
4. SCL – pin 21 (SDL) pada Arduino Mega



**Gambar 3.10** Schematic RTC DS1307 dengan Arduino Mega

Sedangkan untuk pencatatan datanya diperlukan memori untuk menyimpan data yang telah terbaca. Memori yang digunakan adalah *microSD* yang telah terpasang pada modul *microSD Adapter*. Berikut adalah sambungan antara *microSD Adapter* dengan Arduino Mega :

1. CS (*Chip Select*) – Pin 53 (SS) Arduino Mega
2. SCK (*Serial Clock*) – Pin 52 (SCK) Arduino Mega
3. MISO (*Serial Data Out*) – Pin 50 (MISO) Arduino Mega
4. MOSI (*Serial Data In*) – Pin 51 (MOSI) Arduino Mega
5. VCC – Pin 5 Volt Arduino Mega
6. GND – Pin *Ground* Arduino Mega



**Gambar 3.11** Schematic *microSD Adapter* dengan Arduino Mega

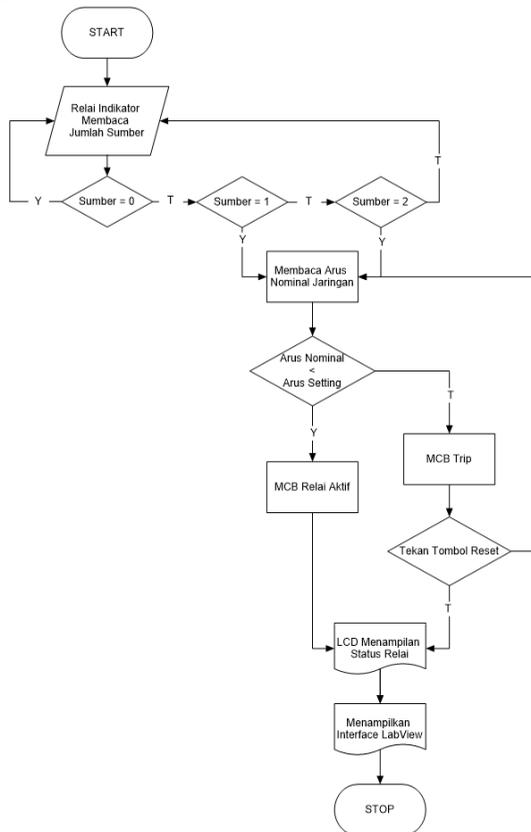
### 3.4 Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak (*software*) menggunakan dua macam *software* yakni Arduino IDE untuk pemrograman pada papan Arduino

serta menjalankan fungsi dari sensor dan perangkat elektronik yang terhubung pada Arduino, sedangkan software LabVIEW digunakan untuk merancang pada sisi *interface* yang nantinya akan menampilkan data yang dikirim oleh Arduino Mega melalui komunikasi serial.

### 3.4.1 Pemrograman *Software* Arduino IDE

*Software* Arduino IDE pada Tugas Akhir digunakan untuk melakukan pemrograman papan Arduino dalam menjalankan sistem secara keseluruhan. *Software* ini menggunakan bahasa pemrograman C. Perancangan dari pemrograman ini seperti yang ditunjukkan pada Gambar 3.11.



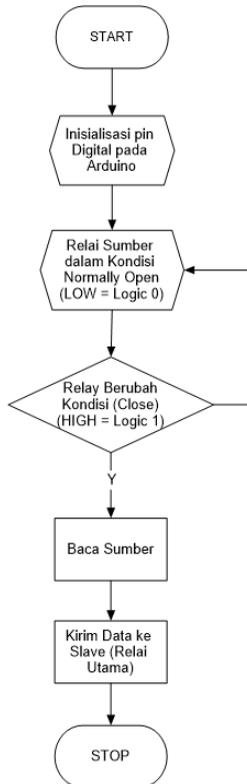
**Gambar 3.12** Flowchart Pemrograman *Software* Keseluruhan Arduino IDE

Dimana ketika program dijalankan relai indikator akan membaca banyaknya sumber. Jika sumber tidak ada maka relai indikator terus membaca hingga sumber terdeteksi. Jika sumber yang melewati jaringan telah terdeteksi maka relai indikator akan mengirimkan kondisi sumber ke relai utama. Selanjutnya relai utama akan membaca data dari relai indikator dan membaca besar arus yang terdapat dalam jaringan. Dari data sumber yang didapat dari relai indikator, relai utama dapat menentukan arus *setting*-nya. Jika arus nominal jaringan kurang dari arus *setting*, maka MCB relai utama akan terhubung. Namun jika arus nominal lebih besar dari arus *setting* maka relai utama akan mendeteksi gangguan arus lebih dan MCB akan *trip* atau memutuskan jaringan. Jika tombol reset relai ditekan maka relai akan membaca arus pada jaringan lagi. Data yang ada pada relai akan ditampilkan pada LCD, disimpan pada data logger dan dikirimkan ke labVIEW sebagai *human interface*.

#### 3.4.1.1 Pemrograman Relai Indikator

Relai indikator ini memiliki fungsi untuk mendeteksi ada atau tidaknya sumber dan berapa jumlah sumber yang mensuplai jaringan. Relai ini sangat penting karena agar adaptif relai dapat berjalan, jumlah sumber yang ada sangatlah berpengaruh pada besar arus *setting* yang ada pada relai utama. Ketika beban bertambah, arus yang lewat pada jaringan menjadi lebih besar. Jika arus yang lewat lebih besar daripada arus *setting* maka relai utama akan *trip*. Agar hal ini tidak terjadi, maka perlu ditambah sumber, agar beban yang ada dapat tersuplai semua. Maka jika sumber ditambah, arus *setting* pada relai utama juga harus bertambah. Berikut adalah penjelasan dari *flowchart* program untuk relai indikator:

- a. *Start*, ketika program mulai berjalan.
- b. Inisialisasi pin Digital pada Arduino yang terhubung pada relai.
- c. Kondisi awal relai indikator adalah *Normally Open* atau terbuka, sehingga *logic* yang diterima adalah *LOW* atau 0.
- d. Jika relai berubah kondisi dari *LOW* menjadi *HIGH* atau 1, berarti relai menjadi *Normally Close*. Sehingga Arduino menerima *logic* 1, ini disebut Arduino *Active High*.
- e. Jika relai tidak berubah kondisi maka relai akan terus *Normally Open* hingga berubah kondisi menjadi *Normally Close*.
- f. Jika relai telah berubah maka arduino akan membaca jumlah sumber yang ada.
- g. Setelah itu arduino akan mengirimkan data jumlah sumber kepada *slave* atau relai utama.



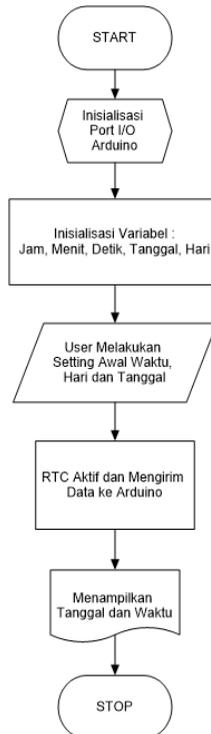
**Gambar 3.13** *Flowchart* Pemrograman Relai Indikator

### 3.4.1.2 Pemrograman RTC

*Real Time Clock* yang digunakan pada Tugas Akhir ini, yaitu Modul *Real Time Clock* DS1307. Untuk pemrograman pada papan Arduino telah tersedia *library* dari RTC tersebut. *Wiring* RTC ke Arduino dengan I<sup>2</sup>C, yaitu menghubungkan pin SDA dan SCL dari RTC ke Arduino pada pin 20 dan 21. Pengguna dapat memanggil *library* yang telah tersedia pada Arduino IDE.

Agar RTC ini dapat digunakan sebagai pemberi data waktu dengan baik maka dibutuhkan program yang sesuai untuk RTC ini. Pada RTC ini set awal untuk data hari, tanggal, serta waktu diberikan pada program dan tidak di set secara manual setelah program diupload. Untuk urutan cara kerja dari *flowchart* adalah sebagai berikut :

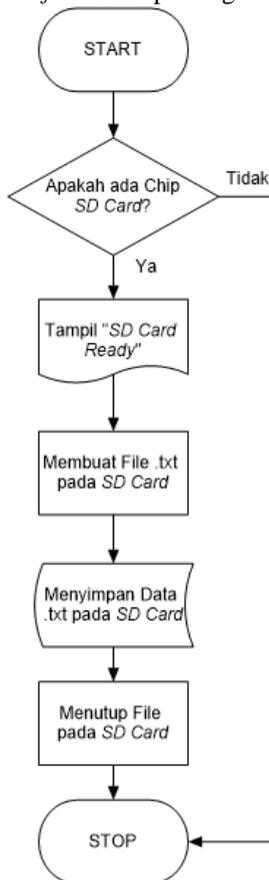
1. *Start* adalah ketika program dimulai.
2. Untuk *wiring* RTC dengan Arduino dihubungkan pada pin SDA dan SCL jadi pada inisialisasi *port I/O* Arduino harus mengaktifkan pin SDA dan SCL. Yakni dengan mengaktifkan komunikasi I<sub>2</sub>C.
3. Selanjutnya inisialisasi variabel yang akan digunakan untuk detik, menit, jam, tanggal, dan hari.
4. Selanjutnya *user* akan melakukan *setting* awal untuk waktu pada RTC tersebut.
5. Dengan data *setting* awal tersebut RTC akan mengirim data tersebut ke Arduino dengan komunikasi I<sub>2</sub>C yakni melalui pin SDA dan SCL.
6. Data waktu akan ditampilkan pada LCD.



**Gambar 3.14** *Flowchart* Pemrograman RTC

### 3.4.1.3 Pemrograman SD Card

Pemrograman data *logger* juga menggunakan pemrograman Arduino IDE. Komunikasi data *logger* berbeda dengan komunikasi dengan RTC yaitu menggunakan SPI. Lebih jelasnya menggunakan pin MISO, MOSI, SCK, dan CS. Pada rancangan *hardware* pin CS terletak pada pin I/O 53. Pin tersebut akan digunakan sebagai acuan untuk mengaktifkan komunikasi dengan relai. Agar *SD Card* ini berjalan dengan baik pada Arduino, maka dibutuhkan pemrograman yang sesuai dengan kebutuhan dari *SD Card*. Berikut adalah *flowchart* pemrograman *SD Card*.



Gambar 3.15 Flowchart Pemrograman SD Card

Untuk urutan cara kerja dari *flowchart* adalah sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Dilakukan pengecekan untuk mengetahui ada atau tidaknya *chip SD Card*.
3. Jika terdapat *chip SD Card* maka akan dimulai untuk proses penyimpanan dari data yang akan disimpan, sedangkan jika tidak ada *chip SD Card* maka program akan berhenti.
4. Tampilan “*SD Card Ready*” untuk proses selanjutnya.
5. Pada tahap ini akan dimulai proses penyimpanan data yang diawali dengan membuka *file* pada *SD Card*.
6. Data akan tersimpan pada *file* yang telah dibuka pada tahap sebelumnya.
7. Setelah data tersimpan, selanjutnya *file* akan ditutup dan data telah selesai tersimpan

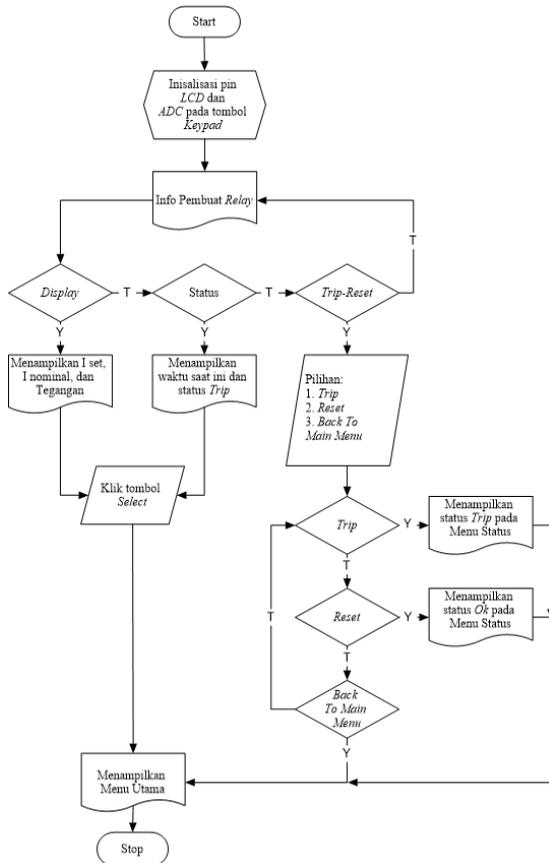
Program tersebut digunakan untuk menulis data *String* pada *file*. Setiap pengaksesan *SD Card* dimulai dengan perintah *SD.open()*; untuk menulis data yang tersimpan dalam memori pada *file* menggunakan *dataFile.print ()*; ketika penulisan selesai maka akan ditutup dengan *dataFile.close()*;

#### **3.4.1.4 Pemrograman LCD**

Pada Tugas Akhir ini *Liquid Crystal Display* atau LCD digunakan untuk mempermudah *user* untuk melihat kondisi terkini dari relai. LCD yang digunakan adalah *LCD Keypad 16 x 2*. Agar *LCD Keypad* ini dapat digunakan sebagai penampil menu kondisi yang terjadi pada relai maka dibutuhkan program yang sesuai untuk *LCD Keypad* ini. Untuk urutan cara kerja dari *flowchart* adalah sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Untuk *wiring LCD Keypad* ini tombol dihubungkan pada Analog 0 Arduino, DB 4 pada Digital 4, DB5 pada digital 5, DB 6 pada Digital 6, DB7 pada Digital 7, pin RS pada Digital 8, *Enable* pada Digital 9 dan *Backlight Control* pada pin Digital 10.
3. Selanjutnya inisialisasi ADC yang akan digunakan untuk mengatur tombol pada *LCD Keypad*.
4. Selanjutnya akan muncul menu utama yang menampilkan info pembuat relai

5. Tombol yang digunakan adalah tombol *Down* untuk menampilkan pilihan menu dan tombol *Select* untuk memilih menu.
6. Menu yang akan ditampilkan ada 3, yaitu :
  - a. Menu *Display* yang berisi *I setting*, *I rms* dan *V rms*
  - b. Menu Status yang berisi Waktu RTC dan Status Jaringan
  - c. Menu *Trip Reset* yang berisi pilihan *Trip*, *Reset* dan *Back to Main Menu*
7. *Stop* adalah ketika program berhenti.



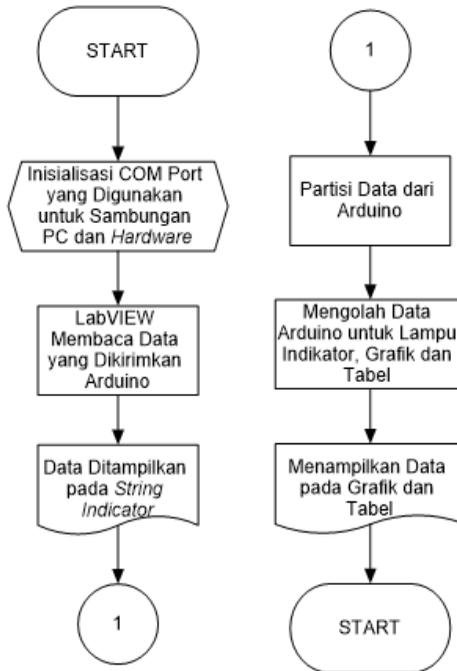
**Gambar 3.16** Flowchart Pemrograman LCD Keypad

### 3.4.2 Pemrograman *Software LabView*

*Software LabView* digunakan sebagai perancangan interface yang nantinya akan menerima data yang dikirim dari papan Arduino. LabVIEW adalah sebuah *software* pemrograman yang diproduksi oleh National Instruments dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau Visual Basic, LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa labVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis text.

Program LabVIEW dikenal dengan sebutan VI atau Virtual Instruments karena penampilan dan operasinya dapat meniru sebuah instrumen. Pada labVIEW, *user* pertama-tama diharuskan membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push buttons*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LED dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu user menyusun blok diagram yang berisi kode-kode VI untuk mengontrol front panel.

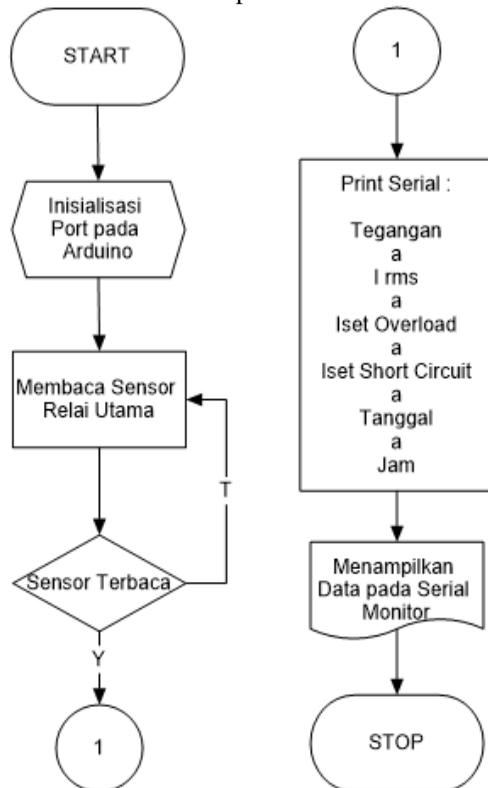
Dari penjelasan tersebut *software LabVIEW* dapat digunakan pada Tugas Akhir ini, yaitu melakukan antar muka dengan program lain serta menampilkan grafik dari data yang dikirim. Komputasi matematik pada *software* ini pun mendukung terhadap pengolahan data untuk menjadi grafik yang nantinya akan ditampilkan pada *personal computer*.



**Gambar 3.17** Flowchart Pemrograman LabVIEW

Perancangan dari *interface* yang akan dibuat pada LabVIEW yakni, pada monitoring adaptif relai data dari arduino akan dikirim ke *personal computer* melalui komunikasi serial. Perlu adanya pengenalan *COM Port* yang digunakan arduino untuk terhubung ke *personal computer*. Ini dilakukan agar data dari arduino dapat diakses oleh LabVIEW. Setelah *COM Port* dapat dikenali, maka LabVIEW akan membaca data yang telah dikirimkan arduino. Setelah itu data akan ditampilkan pada sebuah *string indicator*. Data pada *string indicator* masih berupa data serial arduino dengan format data yang dikirim adalah “Vrms,a,Irms,a,Iset Overload,a,Iset Short Circuit,a,Tanggal,Jam”. Huruf “a” berguna sebagai pemisah tiap data. Agar data dapat diolah lebih lanjut, maka harus dipisahkan terlebih dahulu. Pemisahan ini berdasarkan data sebelum dan sesudah huruf “a”. Setelah data terpisah maka akan ditampilkan ke *string indicator*. Data yang telah dipisah dapat diproses untuk selanjutnya

dijadikan data grafik, tabel dan lampu indikator. Berikut adalah *flowchart* untuk pengiriman data serial arduino pada labview.



**Gambar 3.18** *Flowchart* Data Serial Arduino

-----Halaman ini sengaja dikosongkan-----

## **BAB IV**

### **PENGUJIAN DAN ANALISA DATA**

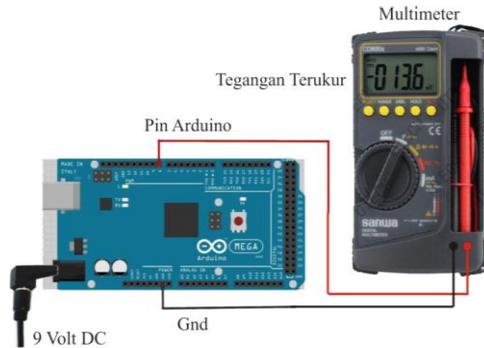
Untuk dapat mengetahui kinerja dari peralatan dan pembuatan sistem yang telah dirancang dan direncanakan sedemikian rupa pada BAB III Perancangan Alat, maka diperlukan pengujian dan analisa data dari setiap komponen pendukung yang dibuat agar sistem dapat berjalan dengan baik sesuai dengan yang diharapkan. Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai pada pengujian sistem. Pengujian juga bertujuan untuk mengetahui kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan atau kesalahan dalam sistem

Pada bab ini akan dibahas tentang pengujian dan analisa data *hardware* dan *software* yang telah dibuat. Adapun bagian – bagian yang akan diuji pada alat ini adalah :

1. *Input/Output* Arduino Uno
2. Pengujian Relai Indikator
3. Pembacaan RTC
4. Memori *SD Card* (*Data Logger*)
5. Pengujian Komunikasi RS485
6. Pengujian *Software* LabVIEW

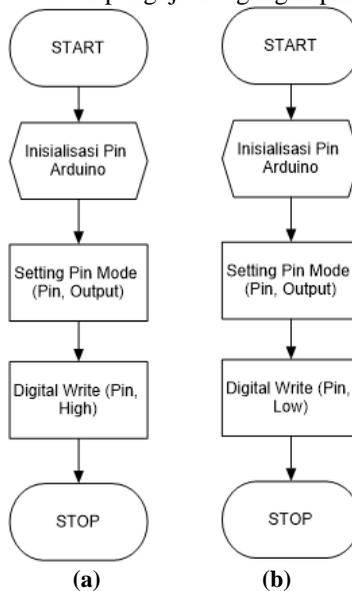
#### **4.1 *Input / Output* Arduino**

Pegujian ini dilakukan terhadap *board* Arduino yang digunakan yakni Arduino Uno untuk relai indikator. Pengujian ini bertujuan untuk mengetahui bahwa kondisi Arduino Uno dapat digunakan dengan baik untuk Tugas Akhir ini. Skema pengujian ditunjukkan pada Gambar 4.1



**Gambar 4.1** Skema Pengujian Pin Input / Output Arduino

Pengujian dilakukan dengan cara memberikan program pada Arduino yakni memberikan perintah *HIGH* dan *LOW* atau logika 0 dan 1 pada setiap pin Arduino yang akan diuji sesuai dengan *flowchart* yang ditunjukkan pada Gambar 4.2, kemudian mengukur besaran tegangan yang keluar dari pin tersebut seperti yang ditunjukkan pada Gambar 4.1. berikut adalah *flowchart* untuk pengujian tegangan pin Arduino Uno.



**Gambar 4.2** *Flowchart* Program Pengujian Pin Arduino (a) *Logic 1*, (b) *Logic 0*

Berikut ini merupakan tabel hasil pengujian pin Arduino Uno untuk relai indikator :

**Tabel 4.1** Pengujian Relai Indikator

<b>NO</b>	<b>NO PIN ARDUINO</b>	<b>LOGIC</b>	<b>TEGANGAN TERUKUR</b>	<b>LOGIC</b>	<b>TEGANGAN TERUKUR</b>
1	A0	1	4.98 V	0	4.0 mv
2	A1	1	4.98 V	0	4.5 mv
3	A2	1	4.98 V	0	5.1 mv
4	A3	1	4.98 V	0	5.3 mv
5	A4	1	4.98 V	0	5.5 mv
6	A5	1	4.98 V	0	5.7 mv
7	D0	1	4.98 V	0	101.6 mv
8	D1	1	4.98 V	0	11.5 mv
9	D2	1	4.98 V	0	7.7 mv
10	D3	1	4.98 V	0	7.0 mv
11	D4	1	4.98 V	0	6.7 mv
12	D5	1	4.98 V	0	2.9 mv
13	D6	1	4.98 V	0	3.1 mv
14	D7	1	4.98 V	0	3.1 mv
15	D8	1	4.98 V	0	3.3 mv
16	D9	1	4.98 V	0	3.1 mv
17	D10	1	4.98 V	0	2.9 mv
18	D11	1	4.98 V	0	2.9 mv
19	D12	1	4.98 V	0	2.5 mv
20	D13	1	4.98 V	0	1.9 mv

V <sub>reff</sub>	5V	4.98 V
	V <sub>IN</sub>	11.38 V
	3.3V	3.277 V

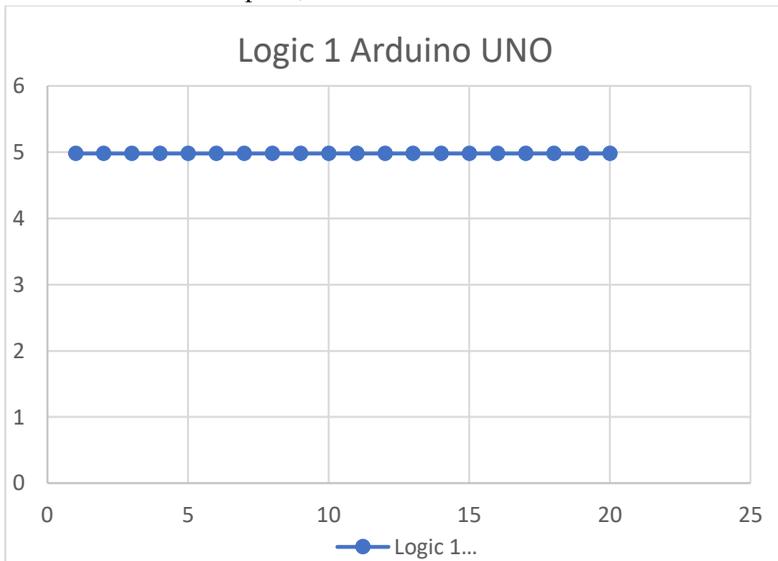
Untuk mengetahui besar error pengukuran data dilakukan penghitungan sebagai berikut ini pada data pin arduino *logic 1*:

$$\text{Nilai Error} = \left| \frac{\text{Nilai Pengukuran} - \text{Nilai Sebenarnya}}{\text{Nilai Sebenarnya}} \right| \%$$

$$\text{Nilai Error} = \left| \frac{4,98 - 5}{5} \right| \%$$

$$\text{Nilai Error} = 0,004\%$$

Dari data tabel diatas dapat disimpulkan dengan kurva garis sebagai berikut. Didapatkan bahwa tegangan pin arduino memiliki kurva yang linear dan error mencapai 0,004 %



**Gambar 4.3** Kurva *Logic 1* Pin Arduino UNO

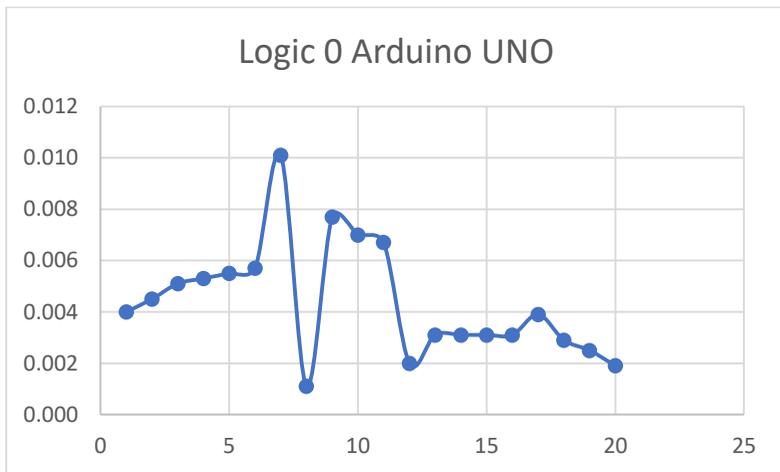
Untuk mengetahui besar error pengukuran data *logic 0* dilakukan penghitungan sebagai berikut:

$$\text{Nilai Error} = \left| \frac{\text{Nilai Pengukuran} - \text{Nilai Sebenarnya}}{\text{Nilai Pengukuran}} \right| \%$$

$$\text{Nilai Error} = \left| \frac{0,004 - 0}{0,004} \right| \%$$
 (Nilai 0,004 dari rata-rata nilai tegangan kondisi *low*)

$$\text{Nilai Error} = 1\%$$

Dari data tabel diatas dapat disimpulkan dengan kurva garis sebagai berikut. Didapatkan bahwa tegangan pin arduino memiliki kurva yang tidak stabil, pada pin D0 dan D1 mengalami kenaikan dan penurunan, karena pin digunakan untuk pengiriman data. Dan error pembacaan mencapai 1%



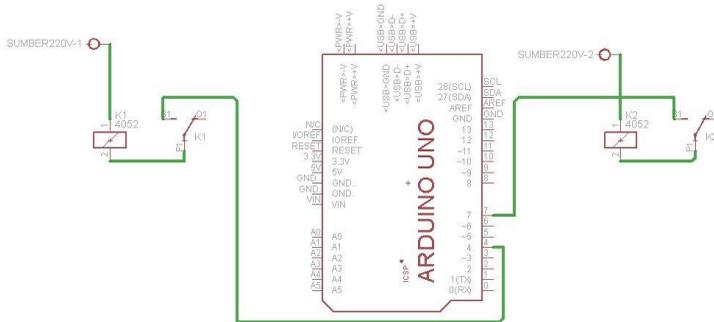
**Gambar 4.4** Kurva *Logic 0* Pin Arduino UNO

Dari data pengujian pin Arduino Uno pada Tabel 4.1 menunjukkan bahwa tegangan keluaran pada setiap pin Arduino konstan sebesar 4,98 Volt ketika Arduino diberi logika 1. Pada keadaan Arduino diberi logika 0 keluaran pada tiap-tiap pin sama yaitu mengeluarkan tegangan bernilai 0 Volt. Hal ini menunjukkan bahwa setiap pin pada Arduino Uno masih bekerja dengan baik dan sesuai dengan datasheet yang ada (*terlampir*).

#### 4.2 Pengujian Relai Indikator

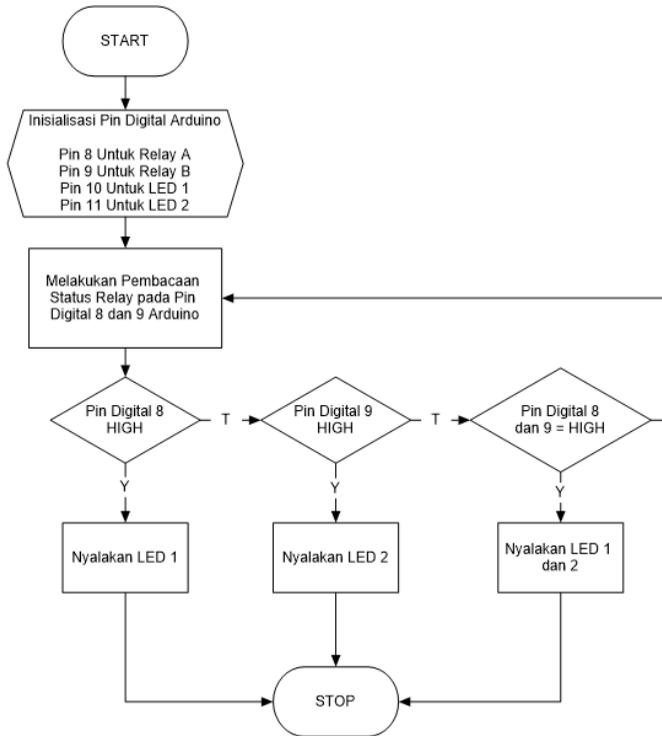
Pengujian relai indikator ini berguna untuk mengetahui apakah alat ini sudah mampu membaca jumlah sumber yang ada. Dengan

memanfaatkan relai AC dan lampu LED, pengujian dilakukan dengan memberikan sumber pada masing-masing relai AC yang selanjutnya akan menyalakan LED jika sumber terdeteksi. Untuk lebih jelasnya berikut adalah skema pengujian relai indikator.



**Gambar 4.5** Schematic Pengujian Relai Indikator

Selain menyusun komponen sesuai dengan skema diatas, dibutuhkan juga program arduino yang sesuai untuk dapat menjalankan relai indikator. Untuk program arduino, pertama adalah menginisialisasi pin digital yang digunakan sebagai *input* dan *output*. Pin 8 dan 9 untuk relai AC (*input*) sedangkan pin 10 dan 11 untuk LED (*output*). Jika sumber 1 yang menyala, maka relai A akan aktif dan mengaktifkan LED 1. Jika sumber 2 yang menyala, maka relai B yang akan aktif dan mengaktifkan LED 2. Dan jika kedua sumber dalam keadaan menyala atau tersambung, kedua relai harus menyala karena kedua relai AC menyala. Untuk lebih jelasnya, berikut ini adalah flowchart pengambilan data untuk relai indikator.



**Gambar 4.6** Flowchart Pengujian Relai Indikator

Setelah melakukan pemasangan komponen dan upload program ke arduino, berikut ini adalah data relai indikator yang telah diambil.

**Tabel 4.2** Pengujian Relai Indikator

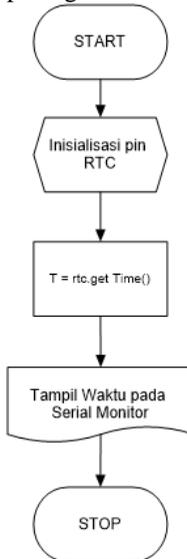
Sumber Aktif	Relai Aktif	LED 1	LED 2
Sumber 1	Relai A	HIGH	LOW
Sumber 2	Relai B	LOW	HIGH
Sumber 1 dan 2	Relai A dan B	HIGH	HIGH

Dari data yang telah didapatkan, dapat dilihat bahwa ketika sumber 1 yang menyala, maka relai A akan aktif dan mengirim perintah Arduino untuk menyalakan LED 1. Ketika sumber 2 yang menyala, maka relai B

akan aktif dan mengirim perintah Arduino untuk menyalakan LED 2. Dan ketika sumber 1 dan sumber 2 sama-sama aktif, maka arduino akan menyalakan kedua LED. Maka relai indikator ini dapat dikatakan sudah siap digunakan untuk mendeteksi jumlah sumber yang ada.

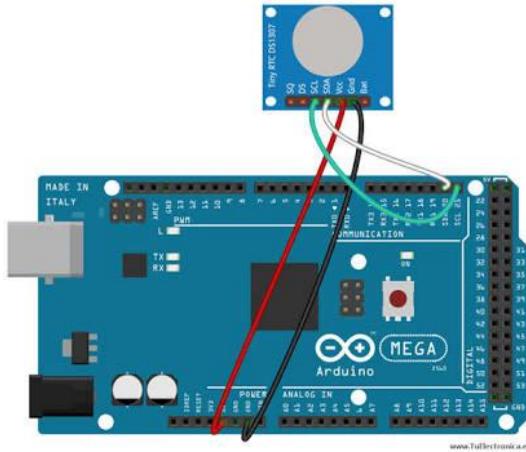
### 4.3 Pembacaan RTC

Pengujian terhadap RTC dilakukan langsung dengan membaca data pada RTC, sebelum dilakukan pengujian maka Arduino diberi program sesuai rancangan *flowchart* pada gambar 4.10 berikut ini.



**Gambar 4.7** Flowchart Pengujian RTC

Setelah program berhasil *upload* ke Arduino Mega, langkah selanjutnya adalah menghubungkan antara Arduino Mega dengan RTC seperti pada skema pengujian yang ditunjukkan pada Gambar 4.11



**Gambar 4.8** Skema Pengujian RTC

Pengujian dilakukan dengan membandingkan tampilan jam pada Serial Monitor Arduino Mega dan tampilan jam pada komputer, yang ditunjukkan pada Gambar 4.12

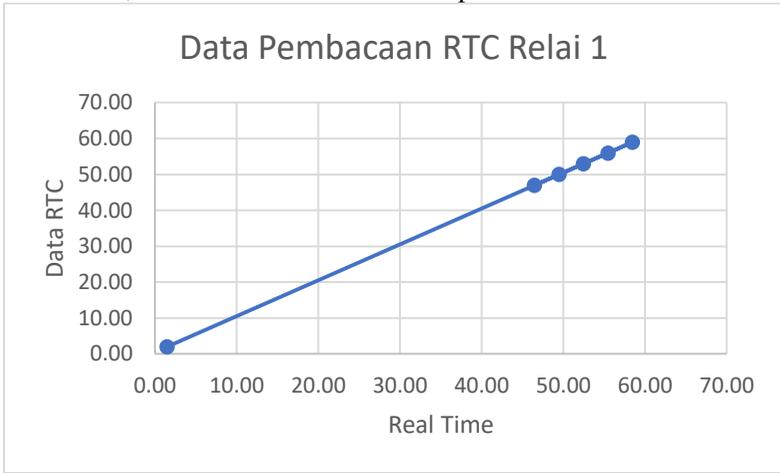
Pengujian ini dibedakan menjadi 2, yaitu pengujian RTC pada relai utama 1 dan relai utama 2. Hasil pengamatan RTC yang telah dibandingkan dengan jam yang terdapat pada komputer ditunjukkan pada tabel dibawah ini :

**Tabel 4.3** Pengujian RTC Relai Utama 1

NO	YANG DIUJI	TAMPILAN SERIAL MONITOR	TAMPILAN KOMPUTER	SELISIH
1	WAKTU	19:46:47	19:47:00	00:00:13
2		19:49:50	19:50:01	00:00:11
3		19:52:48	19:53:00	00:00:12
4		19:55:49	19:56:00	00:00:11
5		19:58:48	19:59:00	00:00:12
6		20:01:48	20:02:00	00:00:12

7	TANGGAL	2017/05/22	22 Mei 2017	-
---	---------	------------	-------------	---

Dari data tabel 4.3 dapat disimpulkan bahwa error pembacaan rtc relai 1 sebesar 0,05% dan berikut adalah kurva pembacaan rtc relai 1 :

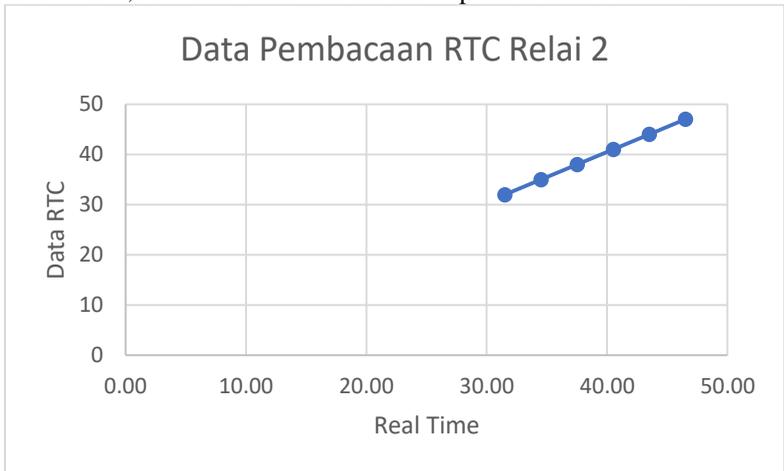


**Gambar 4.9** Kurva Pembacaan RTC Relai 1

**Tabel 4.4** Pengujian RTC Relai Utama 2

NO	YANG DIUJI	TAMPILAN SERIAL MONITOR	TAMPILAN KOMPUTER	SELISIH
1	WAKTU	21:31:51	21:32:00	00:00:09
2		21:34:51	21:35:00	00:00:09
3		21:37:51	21:38:00	00:00:09
4		21:40:51	21:41:00	00:00:09
5		21:43:51	21:44:00	00:00:09
6		21:46:50	21:47:00	00:00:10
7	TANGGAL	2017/05/22	22 Mei 2017	-

Dari data tabel 4.4 dapat disimpulkan bahwa error pembacaan rtc relai 2 sebesar 0,01% dan berikut adalah kurva pembacaan rtc relai 2 :



**Gambar 4.10** Kurva Pembacaan RTC Relai 2

Pada Tabel 4.3 dan 4.4 ditunjukkan hasil pengujian RTC. Hasil pengujian menunjukkan waktu dan tanggal pada RTC telah sesuai dengan tanggal dan waktu pada PC seperti pada Gambar 4.12 dalam 6 kali pengambilan data per detik dengan selisih rata-rata 10 detik, maka dapat disimpulkan bahwa RTC telah sesuai dan dapat digunakan pada sistem.

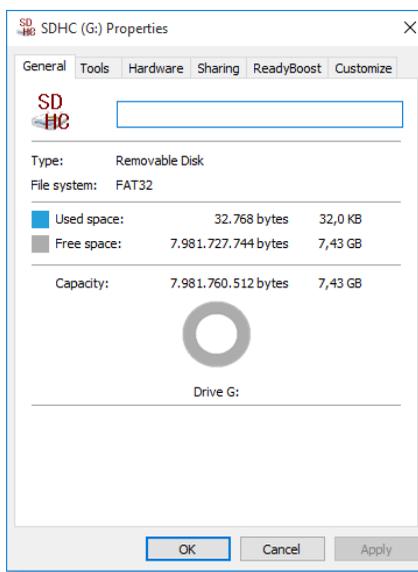
#### **4.4 Memori SD Card (Data Logger)**

Pada pengujian SD Card dilakukan untuk mengetahui kapasitas yang dapat ditampung oleh SD Card. Memori yang digunakan adalah Micro SD Card VGen dengan kapasitas penyimpanan 8 Gb. Pada alat tugas akhir ini memiliki 2 SD Card.

Pengujian dilakukan dengan pembacaan kapasitas kartu pada komputer dalam kondisi kosong. Pengujian dilakukan untuk memastikan SD Card memiliki ruang penyimpanan yang benar-benar kosong, yang nantinya akan diisi data arus dan tegangan. Berikut adalah data SD Card yang telah diambil dan terbagi menjadi beberapa bagian yang seperti pada Tabel 4.5 dan 4.6

**Tabel 4.5** Pengujian SD Card 1

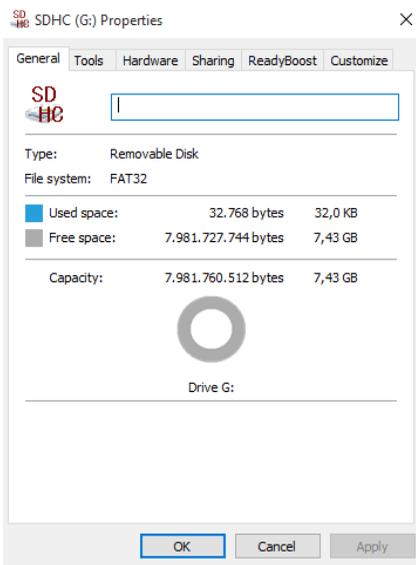
Nama Ruang	Ukuran ( <i>bytes</i> )	Ukuran
<i>Used Space</i>	32.768	32.0 KB
<i>Free Space</i>	7.981.727.744	7.43 GB
<i>Capacity</i>	7.981.760.512	7.43 GB



**Gambar 4.11** Hasil Pengujian SD Card 1

**Tabel 4.6** Pengujian SD Card 2

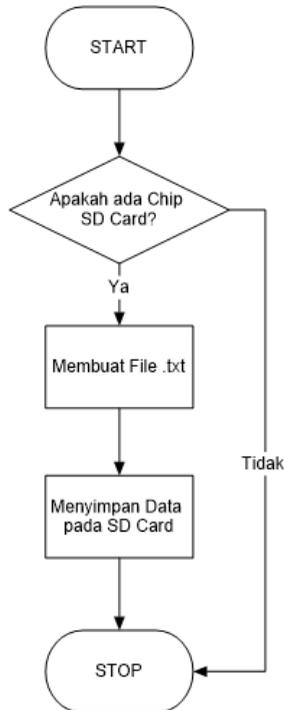
Nama Ruang	Ukuran ( <i>bytes</i> )	Ukuran
<i>Used Space</i>	32.768	32.0 KB
<i>Free Space</i>	7.981.727.744	7.43 GB
<i>Capacity</i>	7.981.760.512	7.43 GB



**Gambar 4.12** Hasil Pengujian SD Card 2

Dengan demikian maka kondisi SD Card dapat digunakan pada Tugas Akhir ini untuk menyimpan nilai arus, dan tegangan pada saat kondisi normal. Selain dengan mengecek memori kartu, dilakukan juga pengujian penyimpanan data arus dan tegangan ke dalam *file text* (.txt). Gambar 4.9 adalah gambar *flowchart* untuk penyimpanan data pada *file text* (.txt). Agar file dapat dijadikan sebuah data yang dapat diproses, maka file .txt diubah kedalam file .xlsx (Microsoft Excel). Langkah langkah untuk mengubah data .txt menjadi .xlsx adalah sebagai berikut :

1. Buka Microsoft Excel
2. Buka menu Data
3. Pada Submenu “*Get External Data*”, pilih “*From Text*”
4. Selanjutnya pilih file .txt penyimpanan data dari Arduino
5. Akan muncul jendela baru “*Text Import Wizard*”, pilih *Delimited* lalu *Next*
6. Lalu pilih *Delimited* (pemisah data) yang digunakan pada *file .txt*, lalu pilih *Next*.
7. Pada data *format* pilih *General*. Tekan *Finish*
8. Pada jendela *Import Data* centang “*Add this data to the Data Model*” lalu klik OK.



**Gambar 4.13** Flowchart Penyimpanan Data .txt pada SD Card

Pengambilan data penyimpanan data dilakukan pada kedua SD Card dengan data adaptif relai. Tabel 4.7 dan 4.8 menunjukkan data yang telah di *import* ke dalam *Microsoft Excel*.

**Tabel 4.7** Penyimpanan Data SD Card 1

No	Tanggal	Waktu	Arus(A)	Tegangan(V)
1	27/05/2017	22:20:04	0,03	209,54
2	27/05/2017	22:20:06	0,03	209,1
3	27/05/2017	22:20:07	0,03	210,09
4	27/05/2017	22:20:09	0,03	209,65
5	27/05/2017	22:20:10	0,03	208,96

6	27/05/2017	22:20:12	0,03	209,75
7	27/05/2017	22:20:13	0,03	209,78
8	27/05/2017	22:20:15	0,04	209,58
9	27/05/2017	22:20:16	0,03	209,29
10	27/05/2017	22:20:17	0,03	209,2
11	27/05/2017	22:20:19	0,04	209,29
12	27/05/2017	22:20:20	0,03	209,28
13	27/05/2017	22:20:22	0,03	209,14
14	27/05/2017	22:20:23	0,03	208,82
15	27/05/2017	22:20:25	0,03	208,82

Gambar 4.10 berikut ini menampilkan data adaptif relai pada LCD, dan telah disimpan di SD Card seperti yang ditunjukkan pada tabel 4.7 yang berwarna kuning.



**Gambar 4.14** Tampilan LCD Penyimpanan SD Card 1

**Tabel 4.8** Penyimpanan Data SD Card 2

No	Tanggal	Waktu	Arus(A)	Tegangan(V)
1	27/05/2017	22:19:03	0,09	60,9
2	27/05/2017	22:19:43	0,06	233,24
3	27/05/2017	22:19:46	0,04	195,49
4	27/05/2017	22:19:48	0,03	193,93
5	27/05/2017	22:19:49	0,03	195,1
6	27/05/2017	22:19:51	0,03	195,15
7	27/05/2017	22:19:52	0,23	194,66

8	27/05/2017	22:19:54	0,25	194,79
9	27/05/2017	22:19:55	0,24	194,52
10	27/05/2017	22:19:56	0,25	194,56
11	27/05/2017	22:19:58	0,24	194,25
12	27/05/2017	22:19:59	0,66	192,9
13	27/05/2017	22:20:01	0,65	192,96
14	27/05/2017	22:20:02	0,66	192,81
15	27/05/2017	22:20:04	0,65	193

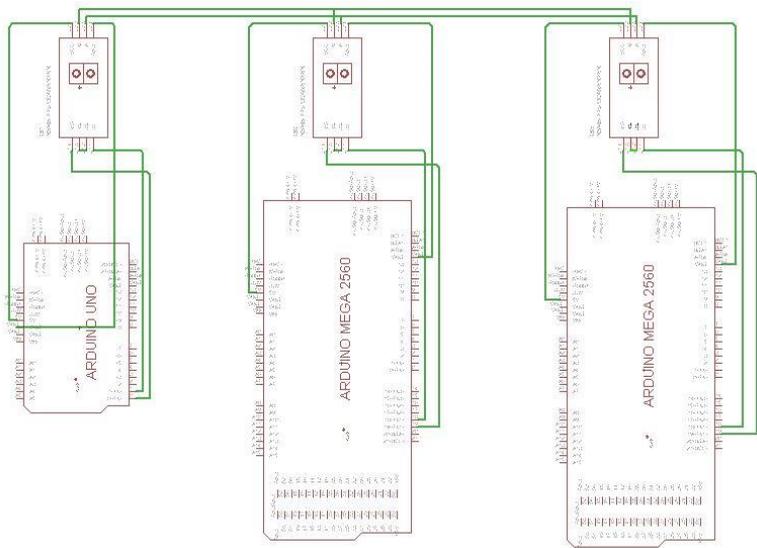
Gambar 4.11 berikut ini menampilkan data arus dan tegangan pada LCD, dan telah disimpan di SD Card seperti yang ditunjukkan pada tabel 4.8 yang berwarna hijau.



**Gambar 4.15** Tampilan LCD Penyimpanan SD Card 2

#### 4.5 Pengujian Komunikasi RS485

Pengujian komunikasi RS485 bertujuan untuk mengetahui apakah relai indikator (*master*) dapat mengirimkan data pada adaptif relai (*slave*), dan sebaliknya. Pengujian ini menggunakan LED sebagai indikator pengiriman data. Arduino *master* dan kedua *slave* dihubungkan dengan RS485. LED pada *master* digunakan sebagai data yang dikirimkan ke *slave*, sedangkan untuk LED indikator dihubungkan pada pin digital tiap-tiap *slave*. Gambar 4.11 menunjukkan skema konfigurasi hubungan Arduino *master* dan *slave*.

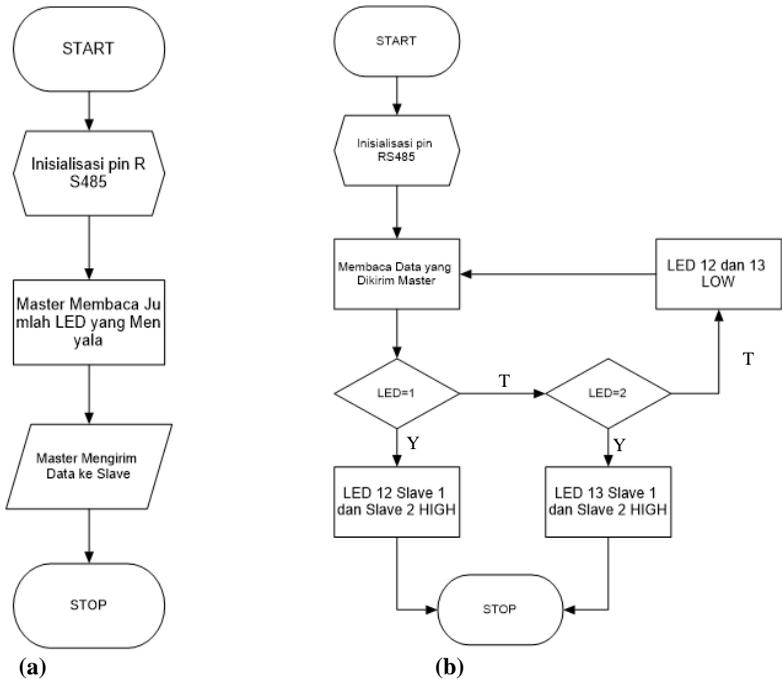


(a)

(b)

**Gambar 4.16** Schematic Komunikasi RS485 (a) Master dan (b) Slave

Program untuk pengujian komunikasi RS485 dapat dilihat pada *flowchart* 4.12. Dimana *master* berfungsi untuk membaca jumlah LED yang menyala dan mengirimkan data ke *slave*. *Slave* menerima data dari master. Jika jumlah LED adalah 1, maka LED hijau dan LED merah akan menyala (LED 12). Sedangkan jika jumlah LED adalah 2, maka LED 13 pada *board* Arduino yang akan menyala. Jika jumlah LED tidak ada, maka LED 12 dan 13 akan tetap mati.



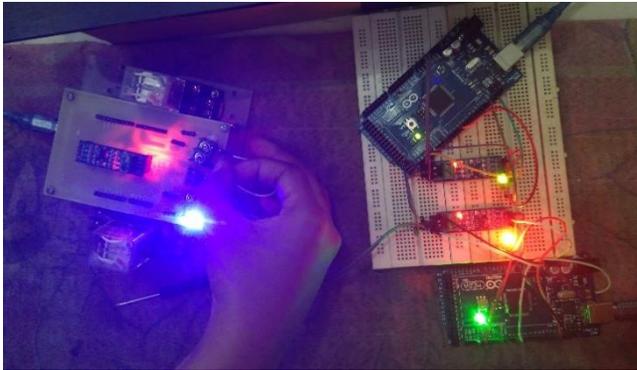
**Gambar 4.17** Flowchart Pengujian Komunikasi RS485 (a) Master dan (b) Slave

Setelah program berhasil *upload* ke *board* Arduino, selanjutnya melakukan pengujian dengan cara menghubungkan LED 1 ke *master*. Setelah itu LED 2 ke *master*, dan yang terakhir kedua LED dihubungkan ke *master*. Tabel 4.9 hasil pengujian pada komunikasi RS485.

**Tabel 4.9** Hasil Pengujian pada Komunikasi RS485

Arduino	Master		Slave 1		Slave 2	
Indikator	LED 1	LED 2	LED 12 (Hijau)	LED 13	LED 12 (Merah)	LED 13
Kondisi	LOW	LOW	LOW	LOW	LOW	LOW
	HIGH	LOW	HIGH	LOW	HIGH	LOW
	LOW	HIGH	LOW	HIGH	LOW	HIGH
	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH

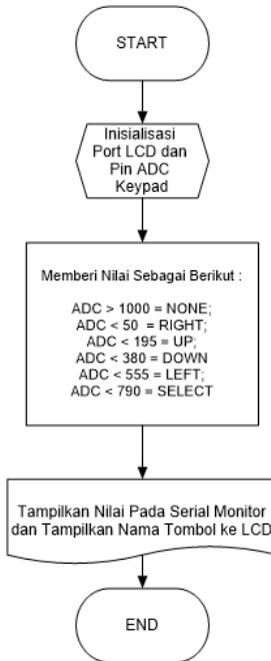
Dari data yang telah didapatkan, komunikasi antara *master slave* Arduino dapat dikatakan berhasil. Karena master dapat mengirimkan data dan *slave* dapat menerima perintah *master* dengan baik, atau error pengiriman sebesar 0%. Berikut adalah gambar proses pengujian komunikasi RS485.



**Gambar 4.18** Proses Pengujian Komunikasi RS485

#### **4.6 Tampilan LCD Keypad 16x2**

Pengujian LCD *Keypad* dilakukan dengan cara menambahkan program pada mikrokontroler Arduino untuk menampilkan karakter pada LCD dan membaca ADC dari tombol *Keypad* ketika ditekan. Pada pengujian LCD kali ini menggunakan LCD *Keypad* 16 x 2, pengujian dilakukan dengan memberi program untuk menampilkan nama tombol *Keypad* yang telah ditekan pada LCD dan menampilkan ADC pada serial monitor. Gambar 4.14 berikut ini adalah *flowchart* pemrograman yang digunakan untuk pengujian LCD pada Adaptif Relai.



**Gambar 4.19** Flowchart Pengujian LCD

Setelah program berhasil *upload* ke *board* Arduin, selanjutnya melakukan penyambungan antara Arduino dengan LCD *Keypad* 16x2. Pada Tabel 2.1 telah dijelaskan konfigurasi penyambungan antara Arduino dengan LCD *Keypad* 16x2. Tabel 4.10 dan 4.11 berikut ini adalah tabel hasil pengujian LCD *Keypad* 16x2 yang dipasang pada kedua adaptif relai.

**Tabel 4.10** Pengujian LCD *Keypad* Relai 1

No	Nilai pada Program	Tampilan LCD	ADC pada Serial Monitor
1	ADC > 1000 NONE	NONE	1023
2	ADC < 50 RIGHT	RIGHT	0
3	ADC < 195	UP	99

	UP		
4	ADC < 380 DOWN	DOWN	255
5	ADC < 555 LEFT	LEFT	407
6	ADC < 790 SELECT	SELECT	638

**Tabel 4.11** Pengujian LCD Keypad Relai 2

No	Nilai pada Program	Tampilan LCD	ADC pada Serial Monitor
1	ADC > 1000 NONE	NONE	1023
2	ADC < 50 RIGHT	RIGHT	0
3	ADC < 195 UP	UP	99
4	ADC < 380 DOWN	DOWN	255
5	ADC < 555 LEFT	LEFT	409
6	ADC < 790 SELECT	SELECT	639

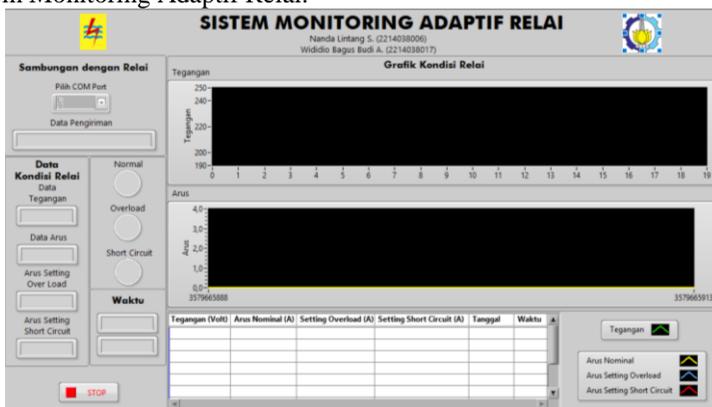
Dari semua data pengujian diatas dapat dilihat bahwa antara tampilan LCD dan ADC yang tampil pada Serial Monitor sesuai dengan nilai yang dimasukkan pada program Arduino. Oleh karena itu, dapat dikatakan bahwa LCD ini dapat berjalan dengan baik pada kedua relai adaptif.

#### 4.7 Pengujian Software LabVIEW

Untuk bisa mengetahui bentuk gelombang dan kondisi terkini dari Adaptif Relai, diperlukan *interface* guna menghubungkan alat kita dengan manusia. *Interface* yang kita gunakan adalah dengan *software* LabVIEW. Pengujian LabVIEW ini digunakan untuk mengetahui apakah LabVIEW sudah bekerja sesuai dengan sistem yang kita inginkan atau belum.

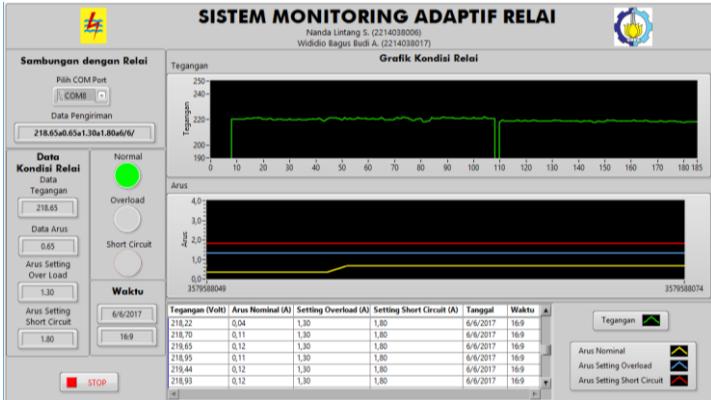
Pada tampilan *interface* Sistem Monitoring Adaptif Relai terdapat beberapa bagian. Yang pertama adalah bagian hubungan dengan

*hardware*, disini berguna untuk memilih *COM Port* hubungan antara *hardware* dan PC. Selain itu juga terdapat tampilan data yang dikirimkan oleh relai. Selanjutnya ada Data Kondisi Relai, dimana terdapat informasi tentang tegangan, arus nominal, arus *setting overload* dan arus *setting short circuit*. Terdapat juga tanggal dan jam pada tampilan *interface* ini. Lalu terdapat grafik tegangan dan arus dari relai. Ada juga indikator lampu yang menandakan bahwa arus normal, *overload* atau *short circuit*. Yang terakhir adalah tabel pencatatan kondisi relai berdasarkan data yang telah dikirim dari arduino ke PC. Berikut adalah tampilan dari *interface* Sistem Monitoring Adaptif Relai.



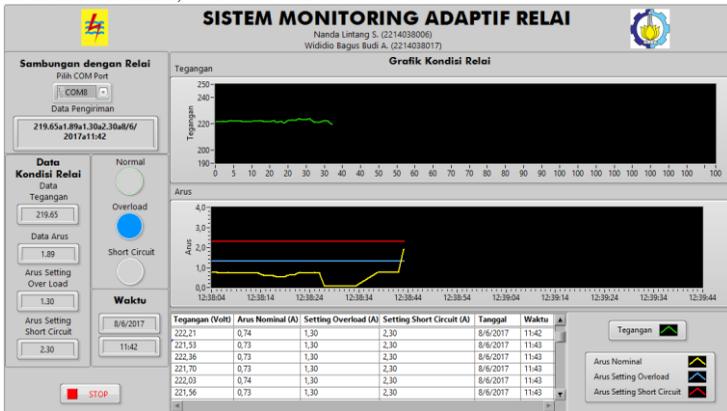
**Gambar 4.20** Tampilan *Interface* Sistem Monitoring Adaptif Relai

Untuk pengujian *interface* Sistem Monitoring Adaptif Relai ini dilakukan dengan memberikan beban pada bus 1 dan bus 2, lalu tiap relai dihubungkan dengan PC. Selanjutnya data pada *interface* dicocokkan dengan data yang tertampil pada relai. Gambar 4. Berikut menunjukkan tampilan *interface* Sistem Monitoring Adaptif Relai ketika relai diberi beban.

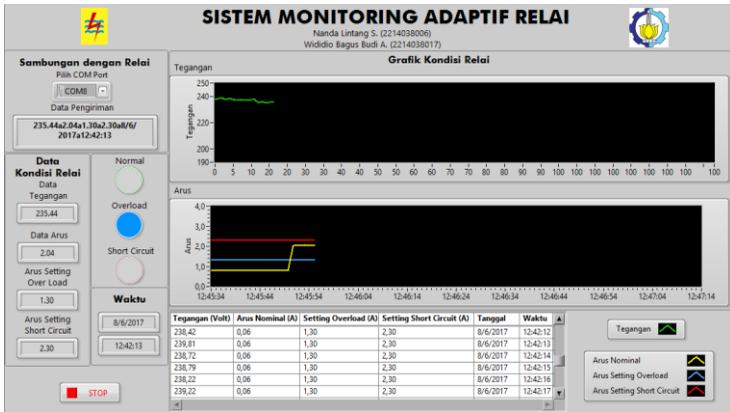


Gambar 4.21 Pengoperasian *Interface* Sistem Monitoring Adaptif Relai

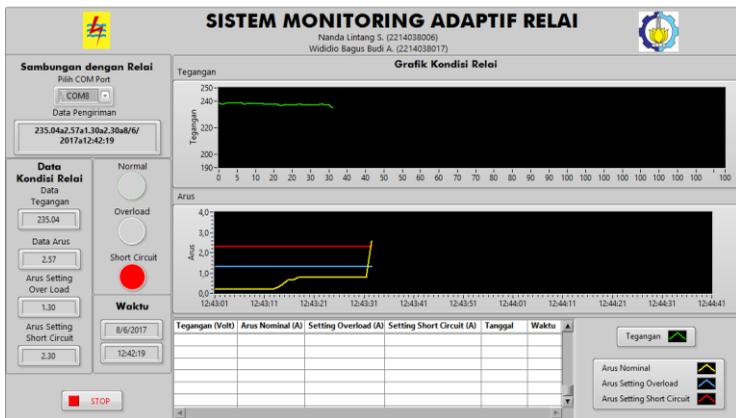
Data yang diambil adalah ketika adaptif relai dalam kondisi normal, *overload* dan *shortcircuit*. Hasilnya adalah apakah *interface* sudah dapat menampilkan data sesuai kondisi adaptif relai. Berikut adalah tampilan *interface* pada kondisi normal, *overload* dan *shortcircuit*.



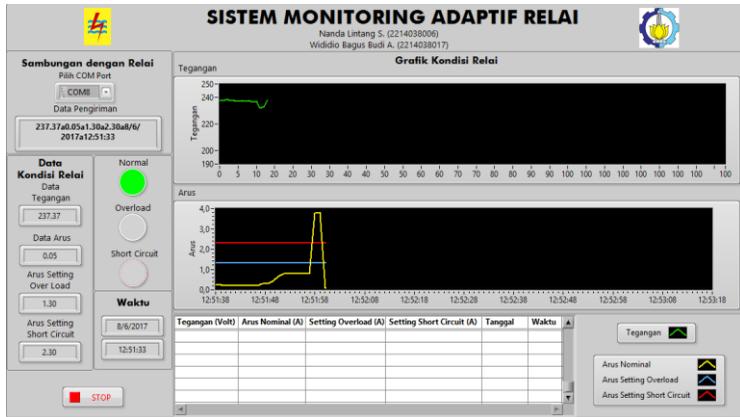
Gambar 4.22 *Interface* Membaca *Overload*



Gambar 4.23 Interface Membaca Overload 2 Detik



Gambar 4.24 Interface Membaca Shortcircuit



Gambar 4.25 Interface Membaca Shortcircuit 1 Detik dan Kembali Normal

#### 4.8 Pengujian Koordinasi Adaptif Relai

Pengujian koordinasi adaptif relai ini bertujuan untuk menguji apakah relai sudah sesuai sistem yang diinginkan. Pengujian ini dilakukan dengan memberikan beban pada tiap-tiap bus dan mencatat kondisi tiap relai. Setting dan kondisi relai dapat dilihat pada table 4.12 dan 4.13. Pengujian keseluruhan dapat dilihat pada Gambar 4.21



Gambar 4.26 Proses Pengujian Koordinasi Adaptif Relai

Tabel 4.12 Hasil Pengujian pada Koordinasi Relai 1 Sumber

1 SUMBER													
KOORDINASI OVERLOAD													
No.	Time Dial Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip		
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2	
1	1	1	1,3	1	0	425 Watt	1,94 A	1,9 A	OK	Trip	-	10,83 s	
2	1	1	1,3	1	425 Watt	0	1,96 A	0,04 A	Trip	OK	16,49 s	-	
3	1	1	1,3	1	300 Watt	100 Watt	1,87 A	0,44 A	Trip	OK	19,93 s	-	
4	1	1	1,3	1	100 Watt	300 Watt	1,88 A	1,37 A	Trip	OK	20,24 s	-	
5	1,2	1	1	1	0	425 Watt	1,94 A	1,88 A	OK	Trip	-	11,88 s	
6	1,2	1	1	1	425 Watt	0	1,94 A	0,04 A	Trip	OK	12,60 s	-	
7	1,2	1	1	1	300 Watt	100 Watt	1,74 A	0,44 A	Trip	OK	13,73 s	-	
8	1,2	1	1	1	100 Watt	300 Watt	1,82 A	1,33 A	Trip	OK	14,69 s	-	
9	1,2	1	1,3	1	100 Watt	300 Watt	1,82 A	1,33 A	OK	Trip	-	23,81 s	
SHORT CIRCUIT													
No.	Time Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip		
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2	
1	2	1	2	2	0	Shortcircuit	2,29 A	2,29 A	OK	Trip	-	1 s	
2	2	1	2	2	Shortcircuit	Load	2,29 A	1,7 A	Trip	OK	2 s	-	
3	1	1	2,3	2	0	Shortcircuit	0	2,2 A	OK	Trip	-	2 Detik	
4	1	1	2,3	2	Shortcircuit	Load	2,5 A	1,5 A	Trip	OK	1 s	-	

Tabel 4.13 Hasil Pengujian pada Koordinasi Relai 2 Sumber



2 SUMBER

KOORDINASI OVERLOAD

No.	Time Dial Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip	
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2
1	1	1	2,3	2	0	575 Watt	2,6 A	2,6 A	OK	Trip	-	27,41 s
2	1	1	2,3	2	575 Watt	0	2,7 A	0,04 A	Trip	OK	43,66 s	-
3	1	1	2,3	2	550 Watt	100 Watt	2,85 A	0,4 A	Trip	OK	33,15 s	-
4	1	1	2,3	2	100 Watt	550 Watt	2,84 A	2,48 A	Trip	OK	32,40 s	-
5	1,2	1	2	2	0	575 Watt	2,54	2,54 A	OK	Trip	-	31,51 s
6	1,2	1	2	2	575 Watt	0	2,57 A	0,03 A	Trip	OK	33,45 s	-
7	1,2	1	2	2	550 Watt	100 Watt	2,9 A	0,44 A	Trip	OK	22,6 s	-
8	1,2	1	2	2	100 Watt	500 Watt	2,91	2,47 A	Trip	OK	22,8 s	-
9	1,2	1	2,3	2	100 Watt	500 Watt	2,9 A	2,47 A	OK	Trip	-	32,94 s

SHORT CIRCUIT

No.	Time Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip	
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2
1	2	1	3	3	0	Shortcircuit	0	3,5 A	OK	Trip	-	1 s
2	2	1	3	3	Shortcircuit	Load	6,1 A	2,9 A	Trip	OK	2 s	-
3	1	1	3,3	3	0	Shortcircuit	3,56	3,56	Trip	Trip	1 s	1 s
4	1	1	3,3	3	Shortcircuit	Load	3,5 A	2,5 A	Trip	OK	1 s	-

Dari data yang telah diambil, dapat dilihat bahwa relai 1 dan relai 2 sudah dapat bekerja sesuai *setting* arus dan *setting* waktu. Ketika di bus 2 mengalami gangguan, maka relai yang memproteksi jaringan adalah relai 2, sedangkan relai 1 dalam kondisi aktif.

#### **4.9 Analisa Relevansi**

Alat ini merupakan perancangan awal untuk monitoring Adaptif Relai. Apabila akan diimplementasikan pada kehidupan sehari-hari perlu adanya penyesuaian dengan beban yang akan diukur, karena pada alat ini kemampuan pengukuran hanya dibatasi pada beban dengan batas arus maksimal 4 A dan hanya dapat digunakan untuk beban 1 fasa. Untuk melakukan pengukuran pada beban 3 fasa maka perlu dilakukan penambahan komponen pada alat yang telah dibuat pada Tugas Akhir ini. Relai arus lebih banyak digunakan pada jaringan listrik sisi tegangan menengah atau pada gardu induk. Sehingga jika akan diimplementasikan secara langsung, dibutuhkan komponen yang mampu menahan arus yang lebih besar. Proses *monitoring* sudah dapat dilakukan secara terus-menerus, sehingga selama hardware dihubungkan dengan monitor data akan terus terpantau.

Diharapkan dengan adanya alat monitoring adaptif relai maka dapat membantu PT PLN (Persero) dalam memberikan pelayanan yang terbaik kepada konsumen. Serta dapat mencegah adanya kerusakan pada peralatan konsumen karena kesalahan koordinasi yang terjadi pada sistem proteksi.

## **BAB V**

### **PENUTUP**

Setelah melakukan perencanaan dan pembuatan alat, dan telah dilakukan pengujian serta analisa data maka dapat ditarik kesimpulan dan saran dari kegiatan tersebut

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian pada Tugas Akhir ini, maka diperoleh beberapa kesimpulan bahwa :

- Pada pengujian pin arduino memiliki error pengukuran sebesar 0,04% pada kondisi *High* dan 1% pada kondisi *Low*.
- Relai indikator digunakan untuk membaca jumlah sumber yang diwakili oleh jumlah led. Ketika sumber yang ada adalah 1 sumber, led 1 menyala. Ketika sumber yang ada adalah 2 sumber, kedua led menyala.
- Untuk pembacaan RTC didapatkan error pembacaan sebesar 0,05% pada relai 1 dan 0,01% pada relai 2
- Pengiriman data oleh *master* ke *slave* ditandai dengan menyalanya led pada masing-masing *slave*. Ketika *master* mengirim 1, led 12 menyala dan ketika *master* mengirim 2 led 13 menyala.
- Pada monitoring adaptif relai dapat dilihat perubahan yang terjadi pada sumber, arus dan tegangan. Dan monitor memiliki *delay* pembacaan data 2 detik.
- Pada koordinasi relai didapatkan nilai error waktu trip untuk sumber 1 adalah 0,022% pada relai 1 dan 0,048% pada relai 2.
- Dan ketika 2 sumber memiliki error waktu trip sebesar 0,038% pada relai 1 dan 0,063% pada relai 2

#### **5.2 Saran**

Beberapa saran untuk mengembangkan tugas akhir ini adalah bahwa pembacaan kondisi adaptif relai dapat langsung melalui *master*, tidak pada masing-masing *slave*. Sehingga pada masing-masing *slave* dapat mengirimkan kondisi ke *master*. Dalam pengembangan alat ini kedepannya sebaiknya menggunakan genset sebagai sumber tenaga, hal ini diperlukan agar tidak menyebabkan trip MCB pada jaringan. Selain itu dengan menggunakan genset dapat memperbesar batasan arus untuk digunakan

penelitian yang lebih presisi. Dan juga diharapkan alat dapat beroperasi pada jaringan 3 fasa agar dapat menyerupai alat yang ada di lapangan. Dalam pengembangan lebih lanjut kedepanya juga disarankan ditambahkan fitur untuk mendeteksi jenis gangguan lain seperti gangguan fasa ke fasa, 2 fasa ke tanah, dan 3 fasa ke tanah.

## DAFTAR PUSTAKA

- [1] ....., **PENGESAHAN RUPTL PLN TAHUN 2016-2025**, MENTERI ENERGI DAN SUMBER DAYA MINERAL REPUBLIK INDONESIA, 2016.
- [2] Andrianto, Heri; dan Aan Darmawan. 2016. **Arduino Belajar Cepat dan Pemrograman**. Bandung : INFORMATIKA.
- [3] ....., **IDE Software for Arduino**, Datasheet, 2015.
- [4] ....., **Modul Komunikasi RS485**, Datasheet, 2016
- [5] ....., **AZZETTLER RELAYS**, <http://www.azettler.com>, 16 April 2017.
- [6] ....., **OMRON AC RELAYS**, <http://www.ia.omron.com>, 19 Mei 2017.
- [7] Adhitya, W.W. dan Faisal, A., **PERANCANGAN SISTEM MONITORING VOLTAGE FLICKER BERBASIS ARDUINO DENGAN METODE FAST FOURIER TRANSFORM (FFT)**, Tugas Akhir, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2016.
- [8] ....., **INFORCE UNINTERRUPTIBLE POWER SUPPLY 650 VA**, Manual Book, 2009.
- [9] DFROBOT, **ARDUINO LCD KEYPAD DFROBOT**, <https://www.dfrobot.com/>, 19 Mei 2017.
- [10] NATIONAL INSTRUMENT, **LABVIEW 2017**, [www.ni.com](http://www.ni.com), 19 Mei 2017

-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN - A

### 1. Listing Program Relai Indikator (Master)

```
#include <SoftwareSerial.h>
#define SSerialRX    11 //Serial Receive pin
#define SSerialTX    10 //Serial Transmit pin

#define SSerialTxControl 9 //RS485 Direction control

#define RS485Transmit  HIGH
#define RS485Receive  LOW

#define PinLED      13

/*-----( Declare objects )-----*/
SoftwareSerial RS485Serial(SSerialRX, SSerialTX); // RX, TX

const int buttonPin1 = 4; // the number of the pushbutton pin
const int ledPin1 = 5; // the number of the LED pin
const int buttonPin2 = 7; // the number of the pushbutton pin
const int ledPin2 = 6; // the number of the LED pin

// variables will change:
int buttonState1 = 0; // variable for reading the pushbutton status
int buttonState2 = 0;
/*-----( Declare Variables )-----*/
int byteReceived;
int byteSend;
int data;

void setup()
{
  Serial.begin(9600);

  pinMode(PinLED, OUTPUT);
  pinMode(SSerialTxControl, OUTPUT);

  digitalWrite(SSerialTxControl, RS485Receive); // Init
  Transceiver
```

```

// Start the software serial port, to another device
RS485Serial.begin(9600); // set the data rate
// initialize the LED pin as an output:
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
// initialize the pushbutton pin as an input:
pinMode(buttonPin1, INPUT);
pinMode(buttonPin2, INPUT);
}

void loop()
{
buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
// check if the pushbutton is pressed.
// if it is, the buttonState is HIGH:

if (buttonState1 == HIGH&&buttonState2 == LOW)
{
digitalWrite(ledPin1, HIGH);
delay(1000);
digitalWrite(SSerialTxControl, RS485Transmit); // Enable
RS485 Transmit
byteSend = RS485Serial.write("a"); // Send byte to Remote
Arduino

digitalWrite(PinLED, LOW); // Show activity
digitalWrite(SSerialTxControl, RS485Receive); // Disable
RS485 Transmit
}
else {
// turn LED off:
digitalWrite(ledPin1, LOW);
}
if (buttonState2 == HIGH&&buttonState1 == LOW) {
// turn LED on:
digitalWrite(ledPin2, HIGH);
delay(1000);
}
}

```

```

    digitalWrite(SSerialTxControl, RS485Transmit); // Enable
RS485 Transmit
    byteSend = RS485Serial.write("a"); // Send byte to Remote
Arduino

    digitalWrite(PinLED, LOW); // Show activity
    digitalWrite(SSerialTxControl, RS485Receive); // Disable
RS485
}
else {
    // turn LED off:
    digitalWrite(ledPin2, LOW);
}
if (buttonState1 == HIGH && buttonState2 == HIGH) {
    // turn LED on:
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, HIGH);
    delay(1000);
    digitalWrite(SSerialTxControl, RS485Transmit); // Enable
RS485 Transmit
    byteSend = RS485Serial.write("b"); // Send byte to Remote
Arduino

    digitalWrite(PinLED, LOW); // Show activity
    digitalWrite(SSerialTxControl, RS485Receive); // Disable
RS485
}
else {
    // turn LED off:
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
}
}
}

```

## 2. Listing Program Relai Utama (Slave)

```

#if ARDUINO < 100
#include <WProgram.h>
#else

```

```

#include <Arduino.h>
#endif

void setup (){
  setupRelayShortCirt();
  setupRelayOverLoad();
  setupDefault();
  pinMode(pinCS,OUTPUT);
  Serial.begin(9600);

void setting(){
  if (Serial1.available(>0)
  {
    data = Serial1.read(); // Read the byte
    //Serial.println(data);
    if (data == 97){
    // put your main code here, to run repeatedly:
    setupDefault();
    }
    if (data == 98){
    // put your main code here, to run repeatedly:
    setupDefault2();
    }
    }
    }
    //Kalau Sumber 1
void setupDefault(){
  Ishortcircuit = 2.3;
  Ioverload =1.3;
  /*setup Relay Protection Short Circuit*/
  RelayShortCirt.setTimeChar(DEF_TIME);//Time Characteristic
  //RelayShortCirt.setTimeChar(INST_TIME);//Time Characteristic
  //RelayShortCirt.setTimeChar(INVS_STD);//Time Characteristic
  RelayShortCirt.setTimeDef(1000);
  RelayShortCirt.setSetting(Ishortcircuit);//set setting value
  //RelayShortCirt.setTMS(1);//set setting value
  /*setup Relay Protection Over Load (Tambahan 20/05/2017)*/
  //RelayOverLoad.setTimeChar(DEF_TIME);//Time Characteristic

```

```

//RelayOverLoad.setTimeChar(INST_TIME);//Time
Characteristic
RelayOverLoad.setTimeChar(INVS_STD);//Time Characteristic
//RelayOverLoad.setTimeDef(2000);
RelayOverLoad.setSetting(Ioverload);//set setting value
RelayOverLoad.setTMS(1200);//set setting value

}
//Kalau Sumber 2
void setupDefault2(){
Ishortcircuit = 3.3;
Ioverload =2.3;
/*setup Relay Protection Short Circuit*/
RelayShortCirt.setTimeChar(DEF_TIME);//Time Characteristic
//RelayShortCirt.setTimeChar(INST_TIME);//Time Characteristic
//RelayShortCirt.setTimeChar(INVS_STD);//Time Characteristic
RelayShortCirt.setTimeDef(1000);
RelayShortCirt.setSetting(Ishortcircuit);//set setting value
//RelayShortCirt.setTMS(1);//set setting value
/*setup Relay Protection Over Load (Tambahkan 20/05/2017)*/
//RelayOverLoad.setTimeChar(DEF_TIME);//Time Characteristic
//RelayOverLoad.setTimeChar(INST_TIME);//Time
Characteristic
RelayOverLoad.setTimeChar(INVS_STD);//Time Characteristic
//RelayOverLoad.setTimeDef(2000);
RelayOverLoad.setSetting(Ioverload);//set setting value
RelayOverLoad.setTMS(1200);//set setting value
}

void setupRelayShortCirt(){
//setup Relay Protection Short Circuit
RelayShortCirt.setPinTest(PIN_TRIP_SWITCH);
RelayShortCirt.setPinReset(PIN_RESET_SWITCH);
//RelayShortCirt.setPinLED(LED_SH);
RelayShortCirt.setPinRL(PIN_RELAY_OUT);
RelayShortCirt.setActionChar(OVR_ACTION);
RelayShortCirt.setActive(true);

void setupRelayOverLoad(){

```

```

//setup Relay Protection Over Load
RelayOverLoad.setPinTest(PIN_TRIP_SWITCH);
RelayOverLoad.setPinReset(PIN_RESET_SWITCH);
//RelayOverLoad.setPinLED(LED_SH);
RelayOverLoad.setPinRL(PIN_RELAY_OUT);
RelayOverLoad.setActionChar(OVR_ACTION);
RelayOverLoad.setActive(true);

```

### 3. Listing Program Sensor Arus dan Tegangan

```

#include "EmonLib.h"
EnergyMonitor monitoringArusDanTegangan;

void setup() {
    Serial.begin(9600);
    SetupEmonlib();

void loop() {
    if (SequenceUtama.isMiliSecondEvent()) { //Dijalankan Tiap 500
ms
        //updateCommandFromHost();//check data dari Host dulu dan
transfer sebagian ke operasi
        setting();
        transferFromProcessToHoldingRegsArdu();//
        mainMenuRele();
    }
    if (SequenceUtama.isASecondEvent()){//jalan setiap detik
//Baca Data Dari EmonLib (Letakkan Program dibawah)

monitoringArusDanTegangan.calcVI(CROSSING_MAX,TIME_O
UT_MAX); // Calculate all. No.of half wavelengths (crossings),
time-out
    digitalWrite( PIN_LED_LIVE, digitalRead(PIN_LED_LIVE) ^
1 );
    SaatIni = rtc.now();
    }

void SetupEmonlib(){
// Inialisasi EmonLib Tegangan dan Arus

```

```

    monitoringArusDanTegangan.voltage(1, 240, 1.7); // Voltage:
input pin, calibration, phase_shift
    //monitoringArusDanTegangan.voltage(1, 250, 1.7); // Voltage:
input pin, calibration, phase_shift
    monitoringArusDanTegangan.current(3, 10.42); // Current: input
pin, calibration.
    //monitoringArusDanTegangan.current(3,
9.6808139534883720930232558139535); // Current: input pin,
calibration. (RL1)
    //monitoringArusDanTegangan.current(3,
10.480239018765069713806478666527); // Current: input pin,
calibration. (RL2)
}

```

#### 4. Listing Program MMC (SD Card) dan RTC

```

// Program penyimpanan memori Micro SD pada Tanggal dan
Waktu menggunakan DS1307 RTC terkoneksi dalam I2C dan
Wire lib
#include <Wire.h>
#include "RTClib.h"
#include <SD.h>

File myFile; //Pendeklarasian File
RTC_DS1307 rtc; //Pendeklarasian RTC 1307

char Time[20]; //Pendeklarasian Waktu
char Date[20]; //Pendeklarasian Tanggal
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};
int pinCS = 53; // Gunakan pin 10 pada Arduino Uno atau pin 53
pada Arduino Mega

void setup () {

    Serial.begin(9600);
    pinMode(pinCS, OUTPUT);
    // Inisialisasi SD Card
    if (SD.begin())
    {

```

```

    Serial.println("SD card is ready to use.");
} else
{
    Serial.println("SD card initialization failed");
    return;
}
// Inisialisasi RTC
if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
}

if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    // Baris berikut menyetel RTC ke tanggal & waktu sketsa ini
    // disusun
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // Baris ini menetapkan RTC dengan tanggal & waktu yang jelas,
    // misalnya
    // January 21, 2014 pada 3am kamu dapat memanggilnya:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
}
}

void loop () {
    DateTime now = rtc.now();
    sprintf(Time, "%02d:%02d:%02d", now.hour(), now.minute(),
now.second());
    sprintf(Date, "%02d/%02d/%02d", now.day(), now.month(),
now.year());
    Serial.print(Time); // print ke serial monitor
    Serial.print("\t");
    Serial.println(Date);

    myFile = SD.open("Coba11.txt", FILE_WRITE); //membuka file
    dan menulis isi filenya
    if (myFile) {
        myFile.print(Time); // print ke file yang ada di micro SD
        myFile.print(",");
    }
}

```

```

    myFile.println(Date);
    myFile.close(); // tutup file
}
// jika file tidak dapat dibuka, print error.
else {
    Serial.println("error opening test.txt");
}
}
delay(1000);
}

```

## 5. Listing Program LCD Keypad

```

#include <LiquidCrystal.h>
#include <phi_interfaces.h>
#include <phi_prompt.h>
byte keypad_type=Analog_keypad;
char mapping[]={'R','U','D','L','S'};
byte pins[]={PIN_KEYPAD};
int values[]={5, 105, 260, 415, 640};
phi_analog_keypads analogKeypad(mapping, pins, values,
BUTTONS_PER_ROW, BUTTONS_PER_COLUMN);
multiple_button_input * keypads[]={&analogKeypad,0};
char up_keys[]={"U"};
char down_keys[]={"D"};
char left_keys[]={"L"};
char right_keys[]={"R"};
char enter_keys[]={"S"};
char
*
function_keys[]={up_keys,down_keys,left_keys,right_keys,enter_
keys};
LiquidCrystal lcd
(LCD_RS,LCD_EN,LCD_D4,LCD_D5,LCD_D6,LCD_D7);
int global_style=109;

void setup() {
    pinMode(pinCS,OUTPUT);
    Serial.begin(9600);
    arduinoInterfaceSetup();//Setup Keypad-LCD
    pinMode(PIN_LED_LIVE, OUTPUT);
    setupMenuReleComb();
}

```

```

    subMenuInfo();}

void arduInterfaceSetup(){
    lcd.begin(LCD_COLUMNS, LCD_ROWS);
    init_phi_prompt(&lcd,keypads,function_keys,
LCD_COLUMNS, LCD_ROWS, '~'); }
//MenuReleComb
const char ReleCombMn0[] PROGMEM="Trip-Reset"/;
const char ReleCombMn1[] PROGMEM="Display";
const char ReleCombMn2[] PROGMEM="Status";
const char ReleCombMn3[] PROGMEM="Info";
const char* const ReleCombMnItems[] PROGMEM =
{ReleCombMn0, ReleCombMn1, ReleCombMn2,
ReleCombMn3};
phi_prompt_struct mainMenu;

void mainMenuRele(){
    switch (menuId.mainMenu)
    {
        case MAIN_MENU_IDX:
            menuReleComb();
            break;
        case TRIP_RESET_MENU_IDX:
            subMenuTripReset("<<-Trip-Reset->>");
            break;
        case DISPLAY_MENU_IDX:
            subMenuDisPlay();
            break;
        case STATUS_MENU_IDX:
            subMenuStatus();
            break;
        case INFO_MENU_IDX:
            subMenuInfo();
            break;
        default:
            break;
    }
}
}

```

```

void setupMenuReleComb(){
  mainMenu.ptr.list=(char**)&ReleCombMnItems;
  mainMenu.low.i=0;
  mainMenu.high.i=2;
  mainMenu.width=LCD_COLUMNS-
((global_style&phi_prompt_arrow_dot)!=0)-
((global_style&phi_prompt_scroll_bar)!=0);
  mainMenu.step.c_arr[0]=LCD_ROWS-1;
  mainMenu.step.c_arr[1]=1;
  mainMenu.step.c_arr[2]=0
mainMenu.step.c_arr[3]=LCD_COLUMNS-4-
((global_style&phi_prompt_index_list)!=0);  mainMenu.col=0; //
Display menu at column 0
  mainMenu.row=1; // Display menu at row 1
  mainMenu.option=global_style; // Option 0, display classic list,
option 1, display 2X2 list, option 2, display list with index, option
3, display list with index2.
}

```

```

void menuReleComb()
{
  int menu_pointer_1=0; // This stores the menu choice the user
made.
  lcd.clear(); // Refresh menu if a button has been pushed
  center_text("Main Menu");//Menu Title

  select_list(&mainMenu); // Use the select_list to ask the user to
select an item of the list, that is a menu item from your menu.
  menu_pointer_1=mainMenu.low.i; // Get the selected item number
and store it in the menu pointer.
  switch (menu_pointer_1) // See which menu item is selected and
execute that correS_Pond function
  {
    case 0:
      menuId.mainMenu = TRIP_RESET_MENU_IDX;
      menuId.subMenu = 1;
      break;
    case 1:
      menuId.mainMenu = DISPLAY_MENU_IDX;

```

```

    menuId.subMenu = 1;
break;
case 2:
    menuId.mainMenu = STATUS_MENU_IDX;
    menuId.subMenu = 1;
break;
case 3:
    menuId.mainMenu = INFO_MENU_IDX;
    menuId.subMenu = 1;
break;
default:
break;
}
}

void subMenuDisplay(){
    byte currentKey;
    String Sval;
    lcd.clear(); // Refresh menu if a button has been pushed

//Tampilkan pada baris ke 1
    Sval = String("Set:");
    Sval = String(Sval + RelayShortCirt.getSetting());
    Sval = String(Sval + "/");
    Sval = String(Sval + RelayOverLoad.getSetting());//EmonLib
Arus
    Sval = String(Sval + " A");
    lcd.setCursor(0,0);//posisikan kursor pada baris 1 kolom 1
    lcd.print(Sval);

//Tampilkan Nilai Tegangan pada baris ke 2
    Sval = String("V/I:");
    Sval = String(Sval + monitoringArusDanTegangan.Vrms);
//EmonLib Tegangan 09/05/2017
    Sval = String(Sval + "/");
    Sval = String(Sval + RelayShortCirt.getValue());//EmonLib Arus
    lcd.setCursor(0,1);//posisikan kursor pada baris 2 kolom 1
    lcd.print(Sval);

```

```

    currentKey = analogKeypad.getKey(); // Use phi_keypads object
to access the keypad
    switch (currentKey) // See which menu item is selected and execute
that correS_Pond function
    {
        case 'S':
            menuId.mainMenu = MAIN_MENU_IDX;
            menuId.subMenu = 1;
            return;
        break;
        default:
            break;
    }
}

```

```

void subMenuStatus(){
    // menu untuk menampilkan tgl dan waktu pada baris pertama
    // status relay pada baris kedua
    byte currentKey;
    String Sval;
    lcd.clear(); // Refresh menu if a button has been pushed

//Tampilkan pada baris ke 1
//DateTime SaatIni = rtc.now();
Sval = String(SaatIni.day());//ambil tanggal
Sval = String(Sval + "/");
Sval = String(Sval + SaatIni.month());//ambil bulan
Sval = String(Sval + "/");
Sval = String(Sval + (SaatIni.year()-2000));///ambil tahun
Sval = String(Sval + " ");
Sval = String(Sval + SaatIni.hour());///ambil jam
Sval = String(Sval + ":");
Sval = String(Sval + SaatIni.minute());///ambil menit
Sval = String(Sval + ":");
Sval = String(Sval + SaatIni.second());///ambil detik

    lcd.setCursor(0,0);
    lcd.print(Sval);
}

```

```

//Tampilkan status relay pada baris ke 2
lcd.setCursor(0,1);
lcd.print(statusRelay());
currentKey = analogKeypad.getKey(); // Use phi_keypads object
to access the keypad
switch (currentKey) // See which menu item is selected and execute
that correS_Pond function
{
  case 'S':
    menuId.mainMenu = MAIN_MENU_IDX;
    menuId.subMenu = 1;
    return;
  break;
  default:
  break;
}
}

```

```

String statusRelay(){
  byte _state;
  String Sval;
  _state = RelayShortCirt.getState();
  if (_state < RelayOverLoad.getState())_state =
  RelayOverLoad.getState(); //ambil nilai status tertinggi 25/05/2017
  Sval = String("STS:");
  if (_state == STATUS_OK)return (Sval = String(Sval + "OK"));
  else if (_state >= STATUS_TRIP){
    Sval = String(Sval + "TRP->");
    _state = _state - STATUS_TRIP;
  }
  if (_state > STATUS_OK){//Trip atau belum trip dengan beberapa
  status
  //check status Relay
  switch (_state){
    case STATUS_OVER:
      return String(Sval + "OVR");
      break;
    case STATUS_UNDER:
      return String(Sval + "UDR");

```

```

        break;
    case EQL_ACTION:
        return String(Sval + "EQL");
    break;
    case STATUS_TEST_LOCAL:
        return String(Sval + "LOCAL");
    break;
    case STATUS_TEST_KEY:
        return String(Sval + "KEYPAD");
    break;
    case STATUS_TEST_REMOTE:
        return String(Sval + "REMOTE");
    break;
    default:
        break;
    }
}
}

void subMenuInfo(){
char infoMsg[]="by: ..... ";
char buffer[15];
lcd.clear();
lcd.noBlink();
center_text("Protection Relay");// display judul
for (byte i=0;i<strlen(infoMsg);i++)
{
    scroll_text(infoMsg,buffer,14,i-14);
    lcd.setCursor(1,1);
    lcd.print(buffer);
    wait_on_escape(300);
}
menuId.mainMenu = DISPLAY_MENU_IDX;
menuId.subMenu = 1;
}

void subMenuTripReset(char* judulMenu)
{
    byte currentKey;

```

```

lcd.clear(); // Refresh menu if a button has been pushed
center_text(judulMenu);//Menu Title
currentKey = analogKeypad.getKey(); // Use phi_keypads object
to access the keypad
switch (currentKey) // See which menu item is selected and execute
that correS_Pond function
{
    case 'U':
        //menuId.mainMenu = TRIP_RESET_MENU_IDX;
        menuId.subMenu--;
        if (menuId.subMenu < TRIP_MENU_IDX) menuId.subMenu =
BACK_MENU_IDX;//roll over to Back_maneu
        break;
    case 'D':
        menuId.subMenu++;
        if (menuId.subMenu > BACK_MENU_IDX) menuId.subMenu
= TRIP_MENU_IDX;//roll over to Trip_menu
        break;
    case 'S':
        if (menuId.subMenu == TRIP_MENU_IDX){
            menuId.mainMenu = 0;
            RelayShortCirt.setState(STATUS_TEST_KEY);//Test trip
dari keypad
            RelayOverLoad.setState(STATUS_TEST_KEY);//Test trip
dari keypad, 25/05/2017
            return;//ini untuk test trip
        }
        if (menuId.subMenu == RESET_MENU_IDX){
            menuId.mainMenu = 0;
            RelayShortCirt.setReset(true);//Reset dari keypad
            RelayOverLoad.setReset(true);//Reset dari keypad, 25/05/2017
            return;//ini untuk Reset
        }
        if (menuId.subMenu == BACK_MENU_IDX){
            menuId.mainMenu = 0;
            return;
        }
        break;
    default:

```

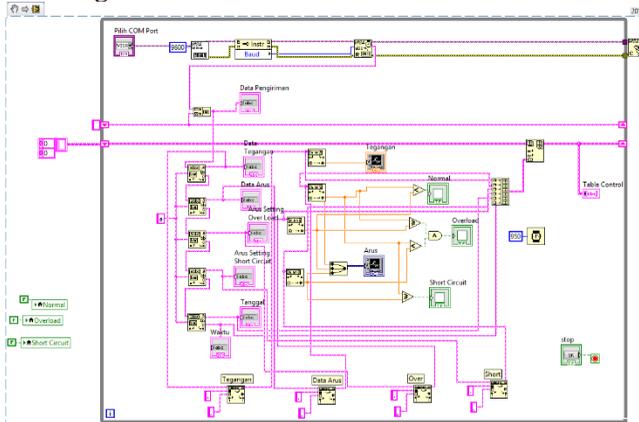
```

        break;
    }
    lcd.clear(); // Refresh menu if a button has been pushed
    center_text(judulMenu);//Menu Title
    DisplayRelayLCD();
}

void DisplayRelayLCD(){
    String Sval;
    byte _timeChar;
    if (menuId.subMenu == TRIP_MENU_IDX){
        Sval = String("Test Trip");
    }
    if (menuId.subMenu == RESET_MENU_IDX){
        Sval = String("Reset");
    }
    if (menuId.subMenu == BACK_MENU_IDX){
        Sval = String("Back To Main Menu");
    }
    lcd.setCursor(0,1);
    lcd.print(Sval);
}

```

## 6. Program LabVIEW



-----Halaman ini sengaja dikosongkan-----



## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an Input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega328P USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I<sup>2</sup>C:** 4 (SDA) and 5 (SCL). Support I<sup>2</sup>C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

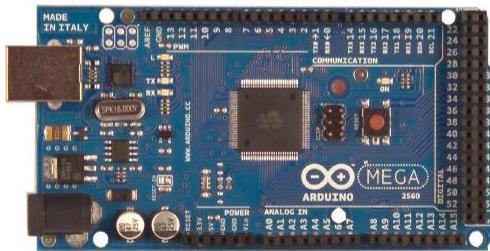
See also the [mapping between Arduino pins and Atmega328 ports](#).

---

## 2. DATASHEET ARDUINO MEGA 2560



### Arduino Mega 2560 Datasheet



### Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

### Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH

---

value, the LED is on, when the pin is LOW, it's off.

- **I<sup>2</sup>C: 20 (SDA) and 21 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I<sup>2</sup>C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I<sup>2</sup>C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

## Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It

communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

### 3. DATASHEET RS485

RS-485 module for Arduino (MAX485)

Click photo above for details, then hover over upper right for more photos.

This module interfaces an Arduino or similar microcomputer to RS-485. RS485 is used for Serial Communications over longer distances than direct RS232 or TTL, and supports multiple units on the same bus (Multi-Drop).

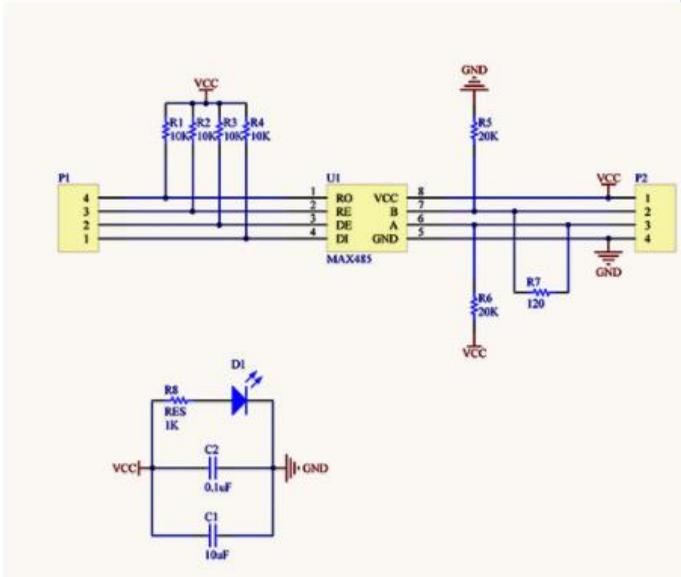
- Multiple Units can be connected to the same RS-485 wiring.
- All Chip pins are brought out for proper controls
- Working voltage: 5V
- Board size: 44 (mm) x14 (mm)

THIS USB INTERFACE for use with PC's is available to connect by RS-485 (click)

How-To Information Link [HERE](#):

See Example RS485 Network Diagram below.

SCHEMATIC OF THIS MODULE:



**Selection Table**

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/DRIVER ENABLE	QUIESCENT CURRENT (µA)	NUMBER OF RECEIVERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8
MAX487	Half	0.25	Yes	Yes	Yes	120	128	8
MAX488	Full	0.25	Yes	No	No	120	32	8
MAX489	Full	0.25	Yes	No	Yes	120	32	14
MAX490	Full	2.5	No	No	No	300	32	8
MAX491	Full	2.5	No	No	Yes	300	32	14
MAX1487	Half	2.5	No	No	Yes	230	128	8

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at [www.maximintegrated.com](http://www.maximintegrated.com).

19-0122; Rev 10, 9/14

## 4. DATASHEET DS1307

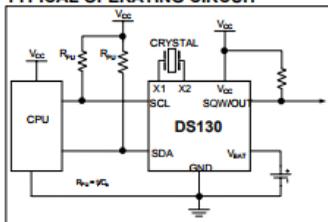


### DS1307 64 x 8, Serial, I<sup>2</sup>C Real-Time Clock

#### GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I<sup>2</sup>C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

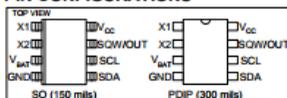
#### TYPICAL OPERATING CIRCUIT



#### FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I<sup>2</sup>C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

#### PIN CONFIGURATIONS



#### ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+ Denotes a lead-free/RoHS-compliant package.

\*A "\*" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.

**ABSOLUTE MAXIMUM RATINGS**

Voltage Range on Any Pin Relative to Ground .....	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial .....	0°C to +70°C
Industrial .....	-40°C to +85°C
Storage Temperature Range .....	-55°C to +125°C
Soldering Temperature (DIP, leads) .....	+260°C for 10 seconds
Soldering Temperature (surface mount) .....	Refer to the JPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

**RECOMMENDED DC OPERATING CONDITIONS**

( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ .) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	$V_{CC}$		4.5	5.0	5.5	V
Logic 1 Input	$V_{IH}$		2.2		$V_{CC} + 0.3$	V
Logic 0 Input	$V_{IL}$		-0.3		+0.8	V
$V_{BAT}$ Battery Voltage	$V_{BAT}$		2.0	3	3.5	V

**DC ELECTRICAL CHARACTERISTICS**

( $V_{CC} = 4.5\text{V}$  to  $5.5\text{V}$ ;  $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ .) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	$I_{LI}$		-1		1	$\mu\text{A}$
I/O Leakage (SDA, SQW/OUT)	$I_{LO}$		-1		1	$\mu\text{A}$
Logic 0 Output ( $I_{OL} = 5\text{mA}$ )	$V_{OL}$				0.4	V
Active Supply Current ( $f_{SCL} = 100\text{kHz}$ )	$I_{CCA}$				1.5	$\text{mA}$
Standby Current	$I_{CCS}$	(Note 3)			200	$\mu\text{A}$
$V_{BAT}$ Leakage Current	$I_{BATLKG}$			5	50	$\text{nA}$
Power-Fail Voltage ( $V_{BAT} = 3.0\text{V}$ )	$V_{PFF}$			$1.216 \times V_{BAT}$	$1.25 \times V_{BAT}$	$1.284 \times V_{BAT}$

**DC ELECTRICAL CHARACTERISTICS**

( $V_{CC} = 0\text{V}$ ,  $V_{BAT} = 3.0\text{V}$ ;  $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ .) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
$V_{BAT}$ Current (OSC ON); SQW/OUT OFF	$I_{BAT1}$			300	500	$\text{nA}$
$V_{BAT}$ Current (OSC ON); SQW/OUT ON (32kHz)	$I_{BAT2}$			480	800	$\text{nA}$
$V_{BAT}$ Data-Retention Current (Oscillator Off)	$I_{BATDR}$			10	100	$\text{nA}$

**WARNING:** Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

**AC ELECTRICAL CHARACTERISTICS**(V<sub>CC</sub> = 4.5V to 5.5V; T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f <sub>SCL</sub>		0	100		kHz
Bus Free Time Between a STOP and START Condition	t <sub>BUF</sub>		4.7			μs
Hold Time (Repeated) START Condition	t <sub>HOLD,STA</sub>	(Note 4)	4.0			μs
LOW Period of SCL Clock	t <sub>LOW</sub>		4.7			μs
HIGH Period of SCL Clock	t <sub>HIGH</sub>		4.0			μs
Setup Time for a Repeated START Condition	t <sub>SUS,STA</sub>		4.7			μs
Data Hold Time	t <sub>HOLD,DAT</sub>		0			μs
Data Setup Time	t <sub>SU,DAT</sub>	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t <sub>tr</sub>				1000	ns
Fall Time of Both SDA and SCL Signals	t <sub>f</sub>				300	ns
Setup Time for STOP Condition	t <sub>SUS,STO</sub>		4.7			μs

**CAPACITANCE**(T<sub>A</sub> = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C <sub>PD</sub>				10	pF
Capacitance Load for Each Bus Line	C <sub>B</sub>	(Note 7)			400	pF

- Note 1:** All voltages are referenced to ground.
- Note 2:** Limits at -40°C are guaranteed by design and are not production tested.
- Note 3:** I<sub>CC3</sub> specified with V<sub>CC</sub> = 5.0V and SDA, SCL = 5.0V.
- Note 4:** After this period, the first clock pulse is generated.
- Note 5:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V<sub>IHQ(DDQ)</sub> of the SCL signal) to bridge the undefined region of the falling edge of SCL.
- Note 6:** The maximum t<sub>SUS,STO</sub> only has to be met if the device does not stretch the LOW period (t<sub>LOW</sub>) of the SCL signal.
- Note 7:** C<sub>B</sub>—total capacitance of one bus line in pF.

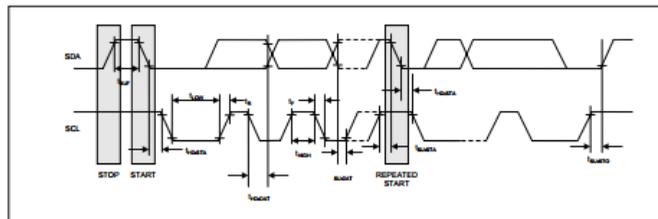
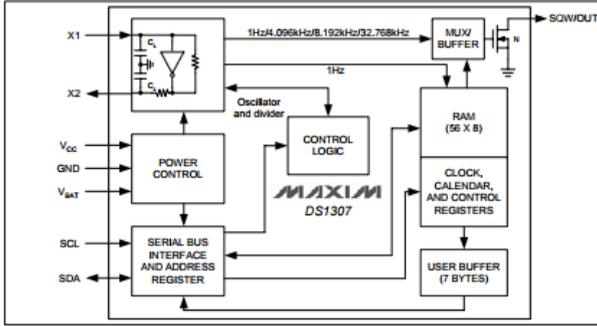
**TIMING DIAGRAM**

Figure 1. Block Diagram



## 5. DATASHEET SCT 13-010



### Product Specification

Date:2015-8-7

Product Name	Current transformer	Model	SCT013-010
--------------	---------------------	-------	------------

Characteristics:Opening size 13mm\*13mm,1m leading wire, standard  $\Phi 3.5$  three-core plug output. Have two kinds of output type: Current output type and voltage output type.  
Purpose: Used for current measurement, monitor and protection for AC motor,lighting equipment, air compressor etc

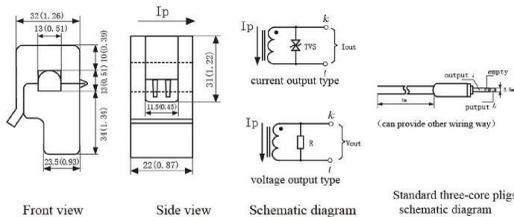
#### Technical Data

$I_{PN}$	Rated input	0-10A
$I_{PM}$	Max. detection input	
$I_{OUT}$	Rated output	0-1V
X	Accuracy	$\pm 1\%$
$E_L$	Linearity	$\leq 0.2\%$
N	Turns ratio	1:1800
$\Phi$	Phase shift	
$R_L$	Max. Sampling resistance	
$V_{PN}$	Work voltage	660V
f	Work frequency	50-1KHz
$T_A$	Operating temperature	-25..+70 $^{\circ}$ C
$T_S$	Storage temperature	-40..+85 $^{\circ}$ C
Vd	Dielectric strength, 50 Hz, 1 min	3KV

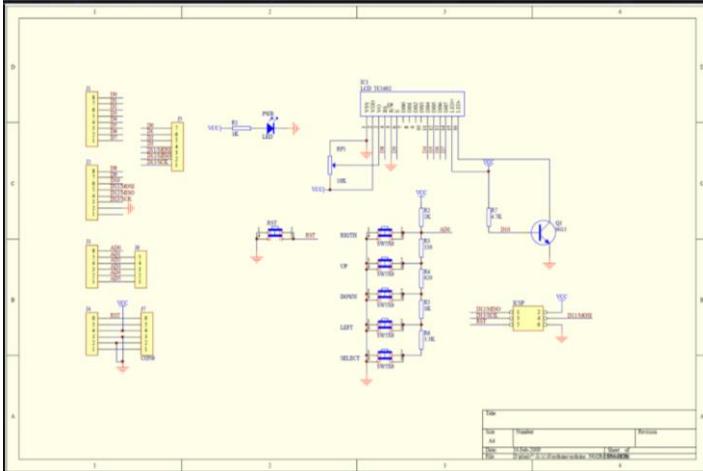


Fire resistance	UL94-V0
Material of core	Ferrite
Mounting type	Suspension
Weight	55g

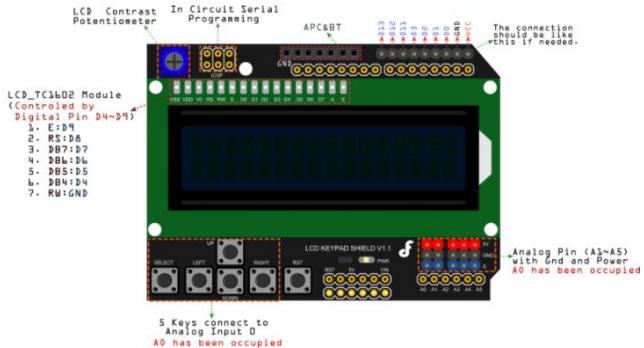
Dimension (mm(in), 1 mm= 0.0394 inch)



## 6. DATASHEET LCD Keypad Shield



### Pinout



#### Instruction for D4 To D10 and Analog Pin 0

Pin	Function	Instruction
Digital 4(D4)		
Digital 5(D5)	D4-D7 are used as	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the LCD.
Digital 6(D6)	DB4-DB7	
Digital 7(D7)		
Digital 8(D8)	RS	Choose Data or Signal Display
Digital 9(D9)	Enable	Starts data read/write
Digital 10(D10)	LCD Backlight Control	
Analog 0(A0)	Button select	Select, up, right, down and left

## 7. Datasheet UPS

**SPESIFIKASI**

MODEL		IF-650 WA	IF-1200 WA
CAPACITY	VA	650VA	1200VA
INPUT	Voltage	110VAC/120VAC or 220VAC / 230VAC /240VAC	
	Voltage Range	81-145VAC or 162-290VAC	
OUTPUT	Voltage Regulation (Batt. Mode)	+/-10%	
	Frequency	50Hz or 60Hz	
	Frequency Regulation (Batt. Mode)	+/-1Hz	
	Output Waveform	Simulated Sine Wave	
BATTERY	Battery Type	12 V/8.2AH x 1	12 V/8.2AH x 2
	Recharge Time	6-8 hours to 90% after complete discharge	
TRANSFER TIME	Typical	2-6 ms	
INDICATOR (*Note 1)	AC Mode	Green LED lighting	
	Battery Mode	Yellow LED Flashing	
	Fault Mode	Red LED Lighting	
AUDIBLE ALARM	Backup Mode	Sounding every 10 seconds	
	Low Battery	Sounding every 1 second	
	Overload	Sounding every 0.5 second	
	Fault	Continuously sounding	
PROTECTION	Full Protection	Discharge, overcharge, and overload protection	
PHYSICAL	Dimension (mm), LXWXH	298x101x142	353x149.3x162
	Operating Environment	0-90% RH @ 0- 40°C (non-condensing)	
ENVIRONMENT	Noise Level	Less than 40dB	

Catatan 1: untuk model LCD silahkan lihat pilihan "3. LCD pada halaman 3

## LAMPIRAN - C

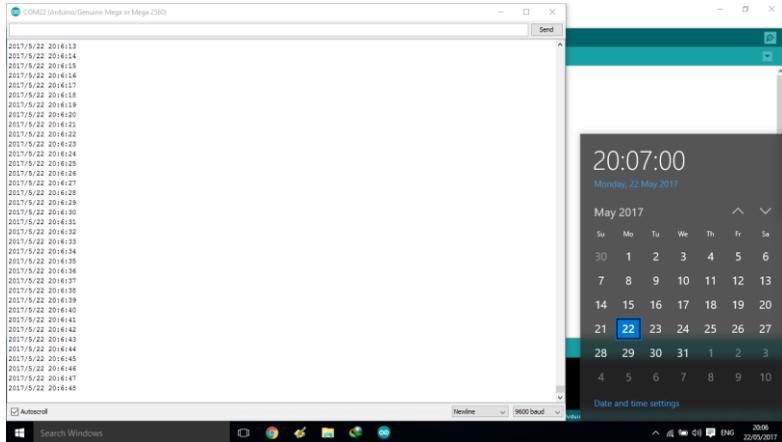
### ALAT ADAPTIF RELAI



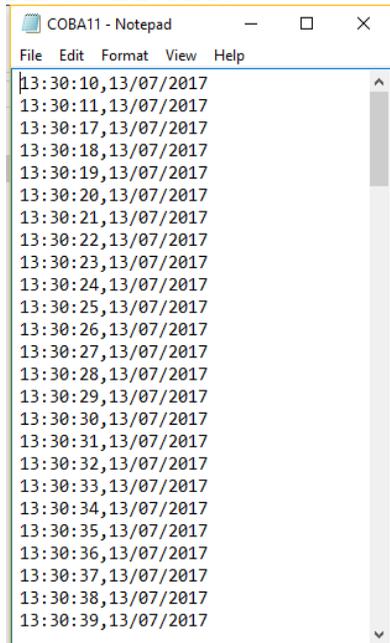
### PENGUJIAN SENSOR ARUS DAN TEGANGAN



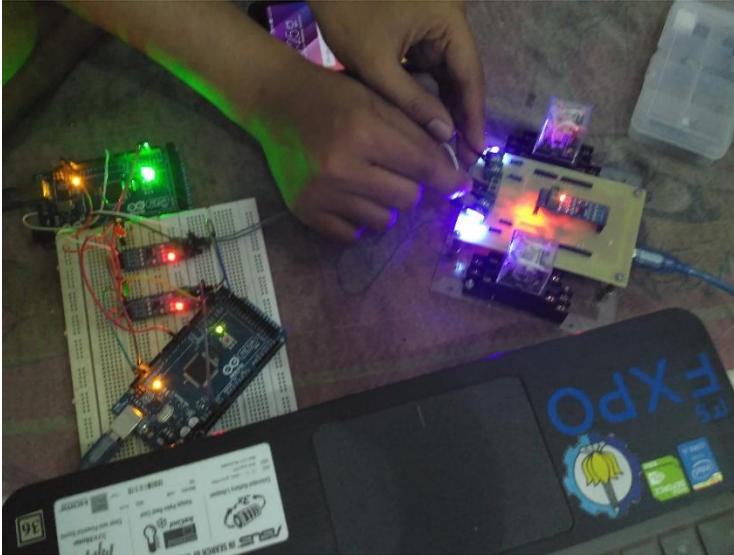
## PENGUJIAN RTC



## PENGUJIAN MMC



## PENGUJIAN RELAY INDIKATOR



## PENGUJIAN RELAY UTAMA





## **RIWAYAT HIDUP PENULIS**



Nama : Nanda Lintang Syafitri  
TTL : Jombang, 30 Juni 1997  
Jenis Kelamin : Perempuan  
Agama : Islam  
Alamat : Desa Betek RT 01 RW 01  
Mojoagung Jombang  
Telp/HP : 085645884887  
E-mail : nandalintang12@gmail.com

### **RIWAYAT PENDIDIKAN**

1. 2002 – 2008 : SDN Betek 1
2. 2008 – 2011 : SMP Negeri 1 Mojoagung
3. 2011 – 2014 : SMA Negeri Jogoroto Jombang
4. 2014 – 2017 : D3 Teknik Elektro Otomasi, Program Studi Teknik Listrik – Fakultas Vokasi Institut Teknologi Sepuluh Nopember (ITS)

### **PENGALAMAN KERJA**

1. Kerja Praktek di PT PLN (Persero) APD Jawa Timur Area Surabaya Selatan

### **PENGALAMAN ORGANISASI**

1. Staff Departemen Big Event HIMAD3TEKTRO 2015-2016
2. Kepala Biro Aplikatif Big Event HIMAD3TEKTRO 2016-2017