



TUGAS AKHIR - TE145561

DATA LOGGER PEMADAMAN GARDU TRAFU TIANG

Regina Yustisia Arifin
Wahyu Hidayanto

NRP 2212 038 014
NRP 2212 038 021

Dosen Pembimbing
Ir. Arif Musthofa, MT.
Rudi Dikairono, ST., MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015

FINAL PROJECT - TE145561

***A LOGGER DATA OF A TRANSFORMER SUBSTATION POLE
BLACKOUT***

Regina Yustisia Arifin
Wahyu Hidayanto

ID 2212 038 014
ID 2212 038 021

Supervisor

Ir. Arif Musthofa, MT.
Rudi Dikairono, ST., MT.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Industrial Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2015

ITS

Institut
Teknologi
Sepuluh Nopember

DATA LOGGER PEMADAMAN GARDU TRAFU TIANG

ITS

Institut
Teknologi
Sepuluh Nopember

ITS

Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Teknik Listrik
Program Studi D3 Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

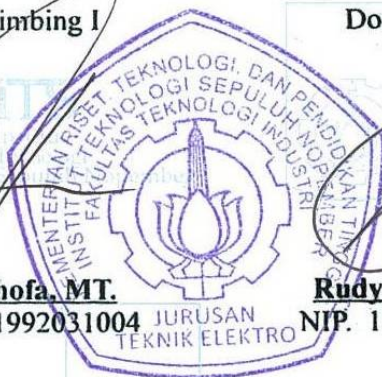
Dosen Pembimbing II

Ir. Arif Musthofa, MT.

NIP. 196608111992031004

Rudy Dikairono, ST., MT.

NIP. 19810325 200501 1 002



SURABAYA
JULI, 2015

DATA LOGGER PEMADAMAN GARDU TRAFU TIANG

Nama Mahasiswa : 1. Regina Yustisia Arifin
2. Wahyu Hidayanto
NRP : 1. 2212 038 014
2. 2212 038 021
Dosen Pembimbing : 1. Ir. Arif Musthofa, MT.
2. Rudi Dikairono, ST., MT.
NIP : 1. 196608111992031004
2. 19810325 200501 1 002

ABSTRAK

PLN (Perusahaan Listrik Negara) yang bertanggung jawab atas listrik di Indonesia, dituntut untuk selalu memberikan pelayanan terbaik untuk konsumen. Salah satu indikasi kualitas pelayanan PLN dapat dilihat dari kontinuitas listrik yang dialirkan ke konsumen.

Kondisi sekarang, laporan padam listrik JTR (jaringan tegangan rendah) didapat bila konsumen menelepon *call centre* 123. Apabila operator telepon pada pelayanan teknik (yantek) telah mendapatkan laporan padam, petugas akan menuju lokasi terjadinya padam listrik. Tentu saja pemadaman yang tidak direncanakan (karena gangguan) ini tidak tercatat secara *real time* karena petugas membutuhkan waktu untuk menelusuri.

Untuk mengatasi hal itu, dibuatlah data logger pemadaman gardu trafo tiang untuk mengetahui status terakhir serta mencatat waktu dan frekuensi padam gardu trafo tiang secara *real time*. Sehingga petugas dapat segera memperbaiki tanpa menunggu adanya laporan padam dari konsumen dan juga dapat mengetahui tingkat keandalan gardu tersebut dalam menyuplai daya ke konsumen.

Alat yang dibuat akan mengukur tegangan *output* trafo distribusi dengan kesalahan pembacaan 2% dan mencatat durasi padam dengan kesalahan sebesar 0.42%. Data padam akan tercatat bila tegangan sama dengan 0 Volt, kemudian dikirim melalui Wi-Fi (*Wireless Fidelity*) ke server. Data yang dikirim berupa waktu, durasi dan frekuensi padam gardu trafo tiang, yang akan digunakan sebagai indikator keandalan gardu dan keandalan PLN dalam menangani gangguan.

Kata Kunci : Data Logger, Gardu Trafo Tiang, Mikrokontroler, Tegangan

Halaman ini sengaja dikosongkan

A LOGGER DATA OF A TRANSFORMER SUBSTATION POLE BLACKOUT

Student's Name : 1. Regina Yustisia Arifin
2. Wahyu Hidayanto
ID : 1. 2212 038 014
2. 2212 038 021
Supervisor : 1. Ir. Arif Musthofa, MT.
2. Rudi Dikairono, ST., MT.
ID : 1. 196608111992031004
2. 19810325 200501 1 002

ABSTRACT

PLN (Perusahaan Listrik Negara) that was responsible to electricity in indonesia , are demanded to always give the best services for consumers . One indication PLN's quality of services can be seen from electrical continuity that distributed to consumers .

The present state, the report of put out electricity in JTR (a low-voltage network) gained if the consumers call the 123. If a telephone operator on the technique (yantek) have received a report put out , the officers will get to the location of the electricity went out. Of course the blackout unplanned (because of strains) it is not recorded in real time because they took some time to track it.

To overcome this, a logger data of a transformer substation pole blackout is made to know the latest status of transformer substation pole and record blackout time and frequency in a real time. so that officers could immediately without waiting for reports from the consumers and can also determine the level of reliability of supply power to consumers .

The device will measure the output voltage distribution transformers with 2% reading error and noting the duration of outages with errors of 0.42%. Data outages will be listed when a voltage equal to 0 Volts, then sent over Wi-Fi (Wireless Fidelity) to the server. The data is sent in the form of time, duration and frequency of the pole transformer substation outages, which will be used as an indicator of the reliability of the substation and the reliability of the PLN in dealing with the disorder.

Key word : logger data, transformer substation pole, microcontroller, voltage

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir berjudul :

Data Logger Pemadaman Gardu Trafo Tiang

Tugas akhir ini merupakan sebagian syarat untuk menyelesaikan mata kuliah dan memperoleh nilai pada tugas akhir.

Dalam proses pembuatan Tugas Akhir ini, penulis mendapat banyak banyak bimbingan serta bantuan dari banyak pihak. Oleh karena itu, dengan tulus ikhlas kami menyampaikan banyak terima kasih kepada:

- a. Kedua orang tua atas limpahan doa, kasih sayang dan perhatian yang telah diberikan kepada penulis
- b. Ir. Arif Musthofa, MT. dan Rudy Dikairono, ST.,MT. selaku dosen pembimbing Tugas Akhir kami, atas segala kesabaran dan kesediaannya meluangkan waktu untuk membimbing serta memberi dukungan sehingga Tugas Akhir ini dapat terselesaikan dengan baik.
- c. Semua teman-teman satu angkatan 2012 D3 Teknik Elektro yang membantu dalam pengerjaan Tugas Akhir ini.
- d. Semua pihak yang telah banyak membantu untuk menyelesaikan Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dalam perancangan dan pembuatan Tugas Akhir ini. Oleh karena itu besar harapan penulis untuk menerima saran dan kritik dari para pembaca. Dan semoga buku Tugas Akhir ini dapat memberikan manfaat bagi para pembaca, khususnya bagi penulis sendiri. Sekian dan terima kasih.

Surabaya, Juni 2015

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Maksud dan Tujuan.....	2
1.5 Metodologi.....	2
1.6 Sistematika Laporan.....	4
1.7 Relevansi.....	4

BAB II TEORI PENUNJANG

2.1 <i>Data Logger</i>	5
2.2 Gardu Trafo Tiang (GTT)	5
2.3 Mikrokontroler ATmega 16	7
2.4 <i>Real Time Clock (RTC)</i>	10
2.5 Sensor Tegangan	12
2.6 RS 232	13
2.7 <i>Wiznet</i>	14
2.8 <i>Wi-Fi</i>	15
2.9 <i>Visual Basic</i>	16

BAB III PERENCANAAN DAN PEMBUATAN

3.1 Perancangan dan Pembuatan <i>Hardware</i>	20
3.1.1 <i>Layout/tata letak</i>	20
3.1.2 Sensor Tegangan.....	21
3.1.3 Mikrokontroler.....	22
3.1.4 <i>RTC</i>	23
3.1.5 <i>Wiznet</i>	24
3.1.6 <i>Router TP Link</i>	27
3.2 Perancangan dan Pembuatan <i>Software</i>	29
3.2.1 <i>CodevisionAVR</i>	29

3.2.2 <i>Visual Basic</i>	33
---------------------------------	----

BAB IV PENGUJIAN DAN PENGUKURAN

4.1 Pengujian Sensor tegangan	35
4.2 Pengujian <i>EEPROM</i>	41
4.3 Pengujian Komunikasi Serial	43
4.4 Pengujian <i>Router Wi-fi</i>	44
4.5 Pengujian <i>Database</i>	46
4.6 Pengujian <i>Visual Basic</i>	47
4.6.1 Pengujian <i>Security</i>	47
4.6.2 Pengujian Tampilan Monitoring	47
4.7 Pengujian Penghitung Lama Waktu Padam	49
4.8 Pengujian Alat Keseluruhan	50
4.8.1 Kondisi Normal Pertama Saat Alat Dinyalakan	51
4.8.2 Kondisi Fasa R Padam	52
4.8.3 Kondisi Fasa S Padam	53
4.8.4 Kondisi Fasa T Padam	54
4.8.5 Kondisi Fasa R dan S Padam	55
4.8.6 Kondisi Fasa R dan T Padam	56
4.8.7 Kondisi Fasa S dan T Padam	57
4.8.8 Kondisi Fasa R, S dan T Padam	58
4.8.9 Kondisi Semua Fasa Kembali Normal	59
4.9 Analisa Relevansi	61

BAB V PENUTUP

5.1 Kesimpulan	63
5.2 Saran	63

DAFTAR PUSTAKA	65
LAMPIRAN 1 LISTING PROGRAM	A-1
LAMPIRAN 2 LISTING DATASHEET	B-1
LAMPIRAN 3 SKEMATIK RANGKAIAN	C-1
RIWAYAT HIDUP PENULIS	D-1

DAFTAR TABEL

Tabel 2.1 Nama dan Fungsi Pin DB9	14
Tabel 2.2 Spesifikasi Modul <i>Wiznet WIZ110SR</i>	15
Tabel 4.1 Hasil Pengujian Sensor Tegangan Fasa R	36
Tabel 4.2 Hasil Pengujian Sensor Tegangan Fasa S	37
Tabel 4.3 Hasil Pengujian Sensor Tegangan Fasa T	38
Tabel 4.4 Hasil Pengujian Sensor <i>Error</i> Tegangan Fasa R	39
Tabel 4.5 Hasil Pengujian Sensor <i>Error</i> Tegangan Fasa S	40
Tabel 4.6 Hasil Pengujian Sensor <i>Error</i> Tegangan Fasa T	40
Tabel 4.7 Rata-rata <i>Error</i> Sensor Tegangan	41
Tabel 4.8 Hasil Pengujian <i>EEPROM</i>	43
Tabel 4.9 Pengujian Penghitung Waktu	49

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 <i>Single Line</i> GTT	6
Gambar 2.2 (a) Gardu Cantol (b) Gardu Portal	7
Gambar 2.3 ATMega 16	8
Gambar 2.4 Nama <i>Port</i> pada AT-Mega 16	9
Gambar 2.5 <i>Real Time Clock</i>	11
Gambar 2.6 Konektor Serial DB-9	13
Gambar 2.7 <i>Wiznet</i>	14
Gambar 2.8 <i>Wi-Fi</i>	16
Gambar 2.9 <i>Layout</i> Pertama <i>Visual Basic 6.0</i>	17
Gambar 3.1 Diagram Fungsional Alat	19
Gambar 3.2 <i>Layout/</i> Tata Letak Alat	20
Gambar 3.3 <i>Wiring</i> pada Pembuatan <i>Hardware</i>	21
Gambar 3.4 Rangkaian Sensor Tegangan	22
Gambar 3.5 Hasil Pembuatan Sensor Tegangan	22
Gambar 3.6 Rangkaian Sistem Minimum	23
Gambar 3.7 Hasil Pembuatan Mikrokontroler	23
Gambar 3.8 Rangkaian RTC dengan IC DS1307	24
Gambar 3.9 Modul <i>Wiznet</i>	25
Gambar 3.10 <i>Configuration Tool</i> WIZ110SR	25
Gambar 3.11 Tampilan Jendela Serial pada <i>Configuration Tool</i> WIZ110SR	26
Gambar 3.12 Pengecekan Koneksi pada <i>Commant Prompt</i>	27
Gambar 3.13 <i>Router</i> TL-MR3020	27
Gambar 3.14 Tampilan <i>Browser</i> Mozilla Firefox	28
Gambar 3.15 Tampilan <i>Network</i> pada Aplikasi TP-LINK	29
Gambar 3.16 <i>Flowchart</i> Mikrokontroler	30
Gambar 3.17 <i>Setting Chip</i> ATMega16	31
Gambar 3.18 <i>Setting</i> I2C	32
Gambar 3.19 <i>Setting</i> Pengiriman Data	32
Gambar 3.20 <i>Setting Input</i> ADC	33
Gambar 3.21 <i>Flowchart Visual Basic</i>	34
Gambar 4.1 Diagram Pengujian Sensor Tegangan	35
Gambar 4.2 Pengujian Sensor Tegangan	36
Gambar 4.3 Grafik Linearisasi Fasa R	37
Gambar 4.4 Grafik Linearisasi Fasa S	37
Gambar 4.5 Grafik Linearisasi Fasa S	38
Gambar 4.6 Skematik <i>EEPROM</i> AT24C32	41
Gambar 4.7 Program <i>CodevisionAVR</i> untuk Pengujian <i>EEPROM</i> ..	42
Gambar 4.8 Hasil Pengujian <i>EEPROM</i>	43

Gambar 4.9 Program <i>CodevisionAVR</i> untuk Pengujian Komunikasi Serial	44
Gambar 4.10 Hasil Tampilan Pengujian Komunikasi Serial	44
Gambar 4.11 Pengujian Koneksi Komputer dengan <i>Router</i>	45
Gambar 4.12 Hasil <i>Ping Test</i> Pengujian <i>Router</i>	45
Gambar 4.13 Tampilan <i>Database</i>	46
Gambar 4.14 <i>Form Login</i> pada HMI	47
Gambar 4.15 Tampilan HMI Data Logger Pemadaman GTT	48
Gambar 4.16 Pengujian Penghitung waktu	49
Gambar 4.17 Kondisi Awal saat Semua Fasa Normal	51
Gambar 4.18 Tampilan Awal HMI saat Semua Fasa Normal	51
Gambar 4.19 Indikator Lampu Fasa R Mati	52
Gambar 4.20 Tampilan HMI Saat Fasa R Padam.	52
Gambar 4.21 Indikator Lampu Fasa S Mati	53
Gambar 4.22 Tampilan HMI saat Fasa S mati	53
Gambar 4.23 Indikator Lampu Fasa T Mati	54
Gambar 4.24 Tampilan HMI Saat Fasa T Padam	54
Gambar 4.25 Indikator Lampu Fasa R dan S Mati	55
Gambar 4.26 Tampilan HMI Saat Fasa R dan S Padam	55
Gambar 4.27 Indikator Lampu Fasa R dan T Mati	56
Gambar 4.28 Tampilan HMI Saat Fasa R dan T Padam	56
Gambar 4.29 Indikator Lampu Fasa S dan T Mati	57
Gambar 4.30 Tampilan HMI Saat Fasa S dan T Padam	57
Gambar 4.31 Indikator Lampu Fasa R, S dan T Mati	58
Gambar 4.32 Tampilan HMI Saat Fasa R, S dan T Padam	58
Gambar 4.33 Indikator Lampu Fasa R, S dan T Kembali Menyala	59
Gambar 4.34 Tampilan HMI Saat Fasa R, S dan T Kembali Menyala.....	59
Gambar 4.35 Total Waktu dan Frekuensi Padam.....	60
Gambar 4.36 Hasil Rekap Data Keterangan Padam.....	61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebagai salah satu aset negara yang bertanggung jawab atas listrik di Indonesia, PLN (Perusahaan Listrik Negara) dituntut untuk selalu memberikan pelayanan terbaik untuk konsumen. Kualitas pelayanan dapat dilihat dari berbagai macam aspek antara lain kualitas daya, kontinuitas pelayanan, dan hal-hal lainnya.

Salah satu indikasi kualitas pelayanan PLN dapat dilihat dari kemampuan PLN mencukupi daya untuk setiap konsumennya. Semakin rendah kontinuitas listrik yang dialirkan, keandalan PLN pun dipertanyakan. Terdapat beberapa jenis ukuran pelayanan untuk PLN dimana diantaranya mengenai SAIDI (*System Average Interruption Duration Index*) dan SAIFI (*System Average Interruption Frequency Index*).

Pendataan mengenai frekuensi padam yang terjadi pada konsumen JTR (Jaringan Tegangan Rendah) selama ini masih dilakukan dengan cara manual yaitu dengan pencatatan waktu padam oleh petugas. Laporan padam listrik di dapat bila konsumen menelepon *call centre* 123. Bila tidak ada laporan, maka PLN tidak akan tau bila terjadi padam. Apabila operator telepon pada pelayanan teknik (yantek) telah mendapatkan laporan padam, petugas yantek akan menuju lokasi penelepon lalu mengecek apakah listrik padam yang terjadi hanya di rumah penelepon saja atau rumah lainnya juga. Apabila rumah lain juga padam, maka petugas akan beralih ke gardu trafo tiang untuk mengecek. Tentu saja pemadaman gardu trafo tiang yang tidak direncanakan (karena gangguan) ini tidak tercatat secara real time karena petugas membutuhkan waktu untuk menelusuri. Selain itu semakin lama respon penanganan padam, maka citra pelayanan PLN akan menurun.

Oleh karenanya, dibutuhkan *data logger* sehingga dapat memonitor sekaligus *merecord* data mengenai waktu terjadi pemadaman secara otomatis. Dengan data yang terekam maka petugas dapat mengetahui status terakhir gardu trafo tiang dan mencatat waktu dan frekuensi padam gardu trafo tiang secara *real time* sehingga petugas dapat segera memperbaiki tanpa menunggu adanya laporan padam dari konsumen dan juga dapat mengetahui tingkat keandalan gardu tersebut dalam menyuplai daya ke konsumen. Pemadaman yang dicatat berupa pemadaman yang disengaja oleh PLN (karena pemeliharaan) serta pemadaman yang tidak disengaja (karena gangguan). Selain itu dengan data durasi dan frekuensi padam yang tersimpan, kita juga dapat

mengitung nilai SAIDI (*System Average Interruption Duration Index*) dan SAIFI (*System Average Interruption Frequency Index*). Namun pada tugas akhir ini kita hanya menyediakan data durasi dan frekuensi padam tanpa menghitung nilai SAIDI (*System Average Interruption Duration Index*) dan SAIFI (*System Average Interruption Frequency Index*).

1.2 Perumusan Masalah

Dalam tugas akhir ini perumusan masalahnya adalah:

1. Pencatatan waktu padam dan nyala gardu trafo tiang yang masih dilakukan secara manual dengan mengandalkan laporan dari masyarakat dan petugas
2. Tidak adanya *data logger* yang dapat menyimpan data waktu dan frekuensi padam gardu trafo tiang
3. Respon penanganan yang lambat terhadap gangguan yang terjadi di gardu trafo tiang disebabkan informasi waktu padam gardu trafo tiang tidak dalam kondisi waktu nyata

1.3 Batasan Masalah

1. Alat “Data Logger Pemadaman Gardu Trafo Tiang” serta komputer server dianggap terus beroperasi (tidak pernah mati)
2. Penyebab terjadinya padam tidak dibahas
3. Alat masih dalam jangkauan *wi-fi* dan jarak *wi-fi* tidak diperhitungkan

1.4 Maksud dan Tujuan

Tujuan penulis menuliskan Tugas Akhir ini adalah:

1. Merancang dan membuat *data logger* yang berfungsi mencatat data waktu dan frekuensi pemadaman listrik pada gardu trafo tiang.
2. Merancang dan membuat sistem metode pengiriman data waktu dan frekuensi pemadaman listrik pada gardu trafo tiang ke server kantor distribusi melalui *Wi-fi (Wireless Fidelity)*.
3. Membuat *interface* berupa tampilan atau monitoring yang dibuat dengan bahasa pemrograman *visual basic* dan program yang mendukung berupa rekapan data-data.

1.5 Metodologi

Dalam pelaksanaan Tugas Akhir yang berupa Data Logger Pemadaman Gardu Trafo Tiang, ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

1. Studi Pustaka dan Survei Data Awal:

Studi Pustaka dan Pembelajaran Kasus : Studi pustaka dilakukan untuk mendapatkan landasan teoritis akan prototipe yang akan dibuat, merumuskan topik dan tujuan konsep kerja, serta mengembangkan gagasan umum yang berhubungan dengan prototipe. Sumber pustaka antara lain dari buku, jurnal, internet, *datasheet*, dan hasil penelitian. Studi pustaka tidak cukup jika digunakan dalam pengembangan dan penerapan ide karena hanya berupa teori saja yang menjadi patokan. Oleh sebab itu, digunakan referensi lain tentang cara pengukuran tegangan pada transformator. Informasi tersebut kami gunakan juga sebagai acuan sekunder dalam pencarian solusi agar proses pencatatan pemadaman gardu trafo tiang dapat dilakukan secara praktis namun tetap menghasilkan data yang valid.

2. Perencanaan dan Pembuatan Alat :

Membuat sistem pencatatan pemadaman dari gardu trafo tiang yang terkoneksi dengan komputer. Dengan memanfaatkan sensor tegangan/arus, dapat diketahui kondisi yang terjadi pada Gardu Trafo Tiang, bahwa Gardu Trafo Tiang tersebut dalam kondisi aktif atau padam. Data yang diperoleh dari sensor tersebut diolah oleh mikrokontroler untuk disesuaikan dengan kondisi *real* di lapangan. Setelah data tersebut diproses oleh mikrokontroler, maka data yang sudah siap dikirim, akan dikirim menggunakan komunikasi *wireless* untuk dihubungkan dengan komputer, komputer menerima data yang dikirim oleh mikrokontroler kemudian data - data pemadaman akan muncul dan diolah melalui komputer oleh operator.

3. Perencanaan dan Pembuatan *Software*:

Dengan merencanakan sebuah perrealisasian sistem dengan membuat *hardware* dan *software* dari sistem dengan pemrograman.

4. Pengujian alat dan Analisis data

Melakukan pengujian atau pengetesan untuk mengetahui apakah rangkaian yang telah dibuat bisa beroperasi dengan baik dan sesuai dengan bahasa pemrograman atau *software* yang dibuat yang kemudian dilakukan penganalisaan data untuk pengecekan.

5. Penyusunan laporan sebagai hasil kesimpulan :

Ketika alat sudah selesai dikerjakan dan sedang dilakukan pengujian maka mulai menyusun laporan akhirnya. Dimana setelah alat sudah siap seratus persen dan telah selesai diuji barulah penyempurnaan laporan akhir dilakukan. Hal ini bertujuan sebagai bentuk pertanggung jawaban dan agar nantinya laporan akhir tersebut dapat menjadi referensi jika ada yang mau mengembangkan alat ini menjadi lebih baik lagi.

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 ataupun studi literatur yang dijadikan sebagai acuan dalam perancangan dan pembuatan alat.

BAB II : PERENCANAAN DAN PEMBUATAN

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 kami buat.

BAB V : PENUTUP

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

1.7 Relevansi

Manfaat Tugas Akhir ini dapat dilihat dari berbagai sisi, antara lain:

1. Sisi PLN
membantu PLN meningkatkan kualitas pelayanan, meminimalisir pengaduan pelanggan akibat padam, serta mempersingkat *response time* karena tidak perlu memeriksa rumah pelanggan yang padam
2. Sisi pelanggan PLN
Mengurangi waktu padam listrik, sehingga masyarakat dapat memanfaatkan energi listrik dengan semestinya.

BAB II

TEORI PENUNJANG

2.1 Data Logger[1]

Logging data (*data logging*) adalah proses otomatis pengumpulan dan perekaman data dari sensor untuk tujuan pengarsipan atau tujuan analisis.

Data logger (perekam data) adalah sebuah alat elektronik yang mencatat data dari waktu ke waktu baik yang terintegrasi dengan sensor dan instrumen didalamnya maupun eksternal sensor dan instrumen. Atau secara singkat *data logger* adalah alat untuk melakukan *data logging*.

Biasanya ukuran fisiknya kecil, bertenaga baterai, portabel, dan dilengkapi dengan mikroprosesor, memori internal untuk menyimpan data dan sensor. Beberapa *data logger* diantarmukakan dengan komputer dan menggunakan *software* untuk mengaktifkan *data logger* dan melihat dan menganalisa data yang terkumpul, sementara yang lain memiliki peralatan antarmuka sendiri (*keypad* dan *LCD*) dan dapat digunakan sebagai perangkat yang berdiri sendiri (*stand-alone device*).

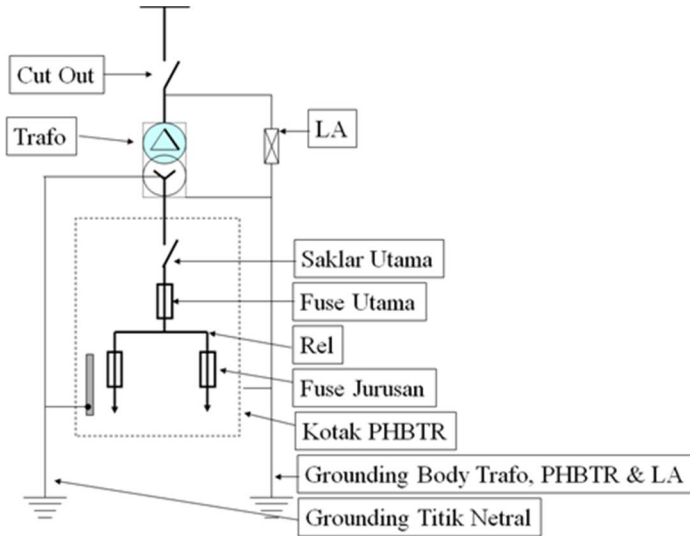
Salah satu keuntungan menggunakan *data logger* adalah kemampuannya secara otomatis mengumpulkan data setiap 24 jam. Setelah diaktifkan, *data logger* digunakan dan ditinggalkan untuk mengukur dan merekam informasi selama periode pemantauan.

2.2 Gardu Trafo Tiang (GTT) [2]

Gardu Trafo Tiang (GTT) adalah merupakan salah satu komponen instalasi tenaga listrik yang terpasang di jaringan distribusi. Berfungsi sebagai trafo daya penurun tegangan dari tegangan menengah ke tegangan rendah, dan selanjutnya tegangan tersebut disalurkan ke konsumen.

Gardu trafo tiang merupakan jenis gardu yang sering digunakan dalam melayani beban, karena biaya instalasinya yang lebih murah dan lebih mudah perawatannya, namun lebih rawan terkena gangguan khususnya gangguan petir.

Mengingat fungsi dan harga trafo tersebut cukup mahal bila dibandingkan dengan peralatan distribusi lainnya, maka pemeliharaan preventif yang dilakukan secara intensif, dengan kriteria pemeliharaan yang jelas untuk setiap komponen GTT dan ditangani oleh tenaga yang terampil dengan peralatan yang memadai agar pemeliharaan tersebut berjalan dengan efektif. Pada gambar 2.1 dapat dilihat *single line* GTT.



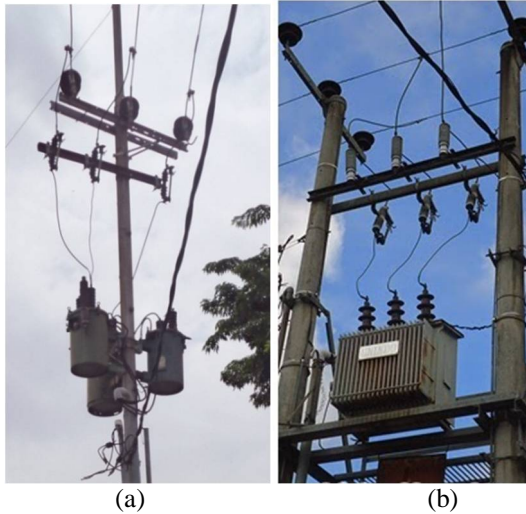
Gambar 2.1 *Single Line GTT*

Untuk mengamankan transformator dan sistemnya, gardu dilengkapi dengan unit-unit pengaman. Karena tegangan yang masih tinggi belum dapat digunakan untuk mencatu beban secara langsung, kecuali pada beban yang didesain khusus, maka digunakan transformator penurun tegangan (*step down*) yang berfungsi untuk menurunkan tegangan menengah 20 KV ke tegangan rendah 400/230 Volt. Gardu trafo distribusi ini terdiri dari dua sisi, yaitu : sisi primer dan sisi sekunder. Sisi primer merupakan saluran yang akan melakukan *supply* ke bagian sisi sekunder. Unit peralatan yang termasuk sisi primer adalah:

- a. Saluran sambungan dari SUTM ke unit transformator.
- b. *Fuse cut out*.
- c. *Ligthning arrester*

GTT merupakan tipe gardu distribusi pasangan luar karena seluruh komponen peralatannya terpasang diluar. Trafo yang terpasang pada GTT berfungsi menurunkan tegangan 20 kV menjadi 380/220 V. Terdapat 2 macam jenis GTT yaitu tipe cantol dan tipe portal. Yang membedakan keduanya adalah jumlah tiang serta kapasitas trafo yang terpasang.

Pada tipe cantol, digunakan 1 buah tiang dan biasanya kapasitas yang terpasang adalah 100 kV. Sedangkan tipe portal menggunakan 2 buah tiang dan biasanya kapasitas trafo terpasang sebesar 160 kV atau lebih. Contoh GTT tipe cantol dapat dilihat pada gambar 2.2 (a) sedangkan tipe portal pada gambar 2.2 (b).



Gambar 2.2 (a) Gardu Cantol (b) Gardu Portal

2.3 Mikrokontroler ATmega 16 [3]

Mikrokontroler adalah mikroprosesor yang dikhususkan untuk instrumentasi dan kendali. Contoh aplikasi pada kendali motor, berperan sebagai PLC (*Programmable Logic Control*) pengaturan pengapian dan injeksi bahan bakar pada kendaraan bermotor, atau alat mengukur suatu besaran, seperti suhu, tekanan, kelembaban dan lain-lain.

Mikrokontroler merupakan suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data.

Mikrokontroler mengerjakan instruksi-instruksi yang diberikan oleh seorang *programmer*. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang *programmer*. Program ini menginstruksikan komputer untuk melakukan aksi-aksi yang sederhana sampai tugas kompleks yang diinginkan *programmer*.

Mikrokontroler merupakan komputer di dalam *chip* yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Mikrokontroler digunakan untuk mengolah data – data *biner* (digital) yang merupakan gabungan dalam bentuk suatu *chip* (IC) serta umunya terdiri dari alamat (*address*), data, pengendali, memori (*RAM* atau *ROM*), dan bagian *input-output*.

AVR merupakan salah satu jenis mikrokontroler yang dibuat oleh Atmel Corp. AVR ini berupa *chip* atau ic yang dapat diprogram karena didalamnya berisi mikroprosesor, memori, dan modul-modul I/O layaknya sebuah komputer. AVR termasuk mikrokontroler Atmel generasi terbaru setelah MCS-51. Teknologi dan fasilitas yang dimiliki AVR menjadikannya lebih terkenal daripada mikrokontroler MCS-51.

Mikrokontroler ATmega16 merupakan salah satu mikrokontroler keluarga AVR. Mikrokontroler ATmega16 memiliki fitur yang cukup lengkap. Mulai dari kapasitas memori program dan memori data yang cukup besar, interupsi, timer/counter, PWM, USART, TWI, analog *comparator*, *EEPROM* internal, dan ADC internal juga terdapat dalam mikrokontroler ATmega16. Sehingga dengan fitur yang lengkap ini memudahkan untuk belajar mikrokontroler keluarga AVR dengan lebih mudah dan efisien. Bentuk fisik Atmega 16 dapat dilihat pada gambar 2.3.

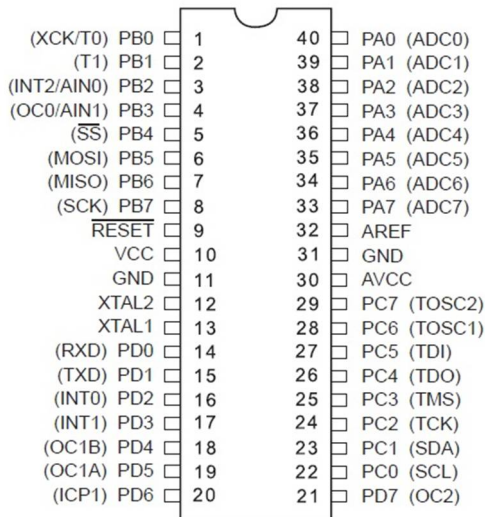


Gambar 2.3 *ATMega 16*

Mikrokontroler ATmega16 memiliki keistimewaan tersendiri dibanding mikrokontroler yang lainnya. Berikut ini adalah fitur-fitur yang dimiliki oleh mikrokontroler ATmega16 :

- a. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah.
- b. Memiliki kapasitas *Flash memory* 16 KByte, *EEPROM* 512 Byte dan *SRAM* 1 KByte.
- c. Saluran I/O ada 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
- d. CPU yang terdiri atas 32 buah register.
- e. Unit interupsi internal dan eksternal.
- f. Port USART untuk komunikasi serial.
- g. Fitur peripheral :
 - Tiga buah *timer / counter* dengan kemampuan perbandingan
 - *Real Time Counter* dengan *Oscillator* tersendiri
 - 4 channel PWM
 - 8 channel, 10-bit ADC
 - *Programmable Serial* USART
 - Antarmuka SPI
- h. *EEPROM* sebesar 512 byte yang dapat diprogram saat operasi
- i. Kecepatan nilai (*speed grades*) 0 - 8 MHz untuk ATMega16L dan 0 - 16 MHz untuk ATMega16

Pada gambar 2.4 ditunjukkan nama-nama tiap *port* pada ATMega16.



Gambar 2.4 Nama *Port* pada AT-Mega 16

Berikut adalah keterangan gambar 2.4 :

- a. **VCC**: Tegangan *supply*
- b. **GND**: *Ground*
- c. **PortA (PA7...PA0)**: *Port* yang berfungsi sebagai *input analog* pada *converter A/D*. Selain itu, sebagai *port I/O 8-bit* dua arah, jika *A/D converter* tidak digunakan. Pin-pin *port* dapat menyediakan resistor *internal pull-up* (untuk masing-masing *bit*).
- d. **PortB (PB7..PB0)**: *Port I/O 8 –bit* dengan resistor *Pull Up internal* tiap pin, *Buffer PortB* mempunyai kapasitas menyerap (*sink*) dan Mencatu (*source*).
- e. **PortC (PC7..PC0)**: *Port I/O 8-bit* ([PC6], PC5...PC0) dengan resistor *pull-up internal* tiap pin. *Buffer portC* mempunyai kapasitas menyerap (*sink*) dan mencatu (*source*).
- f. **PortD (PD7..PD0)**: *Port I/O 8-bit* dengan resistor *Pull-up internal* tiap pin. *Buffer port C* mempunyai kapasitas menyerap (*sink*) dan mencatu (*source*).
- g. **AVcc**: *AVcc* adalah pin tegangan catu untuk *A/D converter*. *AVcc* harus dihubungkan ke *Vcc*, walaupun *ADC* tidak digunakan. Jika *ADC* digunakan , maka *AVcc* harus dihubungkan ke *VCC* melalui “*low pass filter*”.
- h. **AREF**: untuk pin tegangan referensi analog untuk *ADC*.
- i. **Reset**: Sebuah *low level* pulsa yang lebih lama daripada lebar pulsa minimum pada pin ini akan menghasilkan *reset* meskipun *clock* tidak berjalan.
- j. **XTAL1**: *Input inverting* penguat *oscilator* dan *input intenal clock* operasi rangkaian.
- k. **XTAL2**: *Output dari inverting* penguat *oscilator*.

2.4 Real Time Clock (RTC)[4]

Real time clock (RTC) adalah jam elektronik berupa *chip* yang dapat menghitung waktu (mulai detik hingga tahun) dengan akurat dan menjaga/menyimpan data waktu tersebut secara *real time*. Karena jam tersebut bekerja *real time*, maka setelah proses hitung waktu dilakukan output datanya langsung disimpan atau dikirim ke device lain melalui sistem antarmuka.

RTC dilengkapi dengan baterai sebagai pensuplai daya pada *chip*, sehingga jam akan tetap *up-to-date* walaupun komputer dimatikan. RTC dinilai cukup akurat sebagai pewaktu (*timer*) karena menggunakan osilator kristal.

RTC kebanyakan 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.

Pada Tugas Akhir ini, RTC yang dipakai adalah DS1307. DS1307 merupakan *Real-time clock* (RTC) yang dapat menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100. 56-byte, *battery-backed*, RAM *nonvolatile* (NV) RAM untuk penyimpanan. DS1307 merupakan *Real-time clock* (RTC) dengan jalur data paralel yang memiliki antarmuka serial *Two-wire* (I2C), Sinyal luaran gelombang-kotak terprogram (*Programmable squarewave*), Deteksi otomatis kegagalan-daya (*power-fail*) dan rangkaian *switch*, Konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional osilator. Tersedia fitur industri dengan ketahanan suhu: -40°C hingga $+85^{\circ}\text{C}$. Tersedia dalam kemasan 8-pin DIP atau SOIC. Gambar RTC dapat dilihat pada gambar 2.5.



Gambar 2.5 Real Time Clock

Berikut penjelasan pin-pin pada IC DS1307:

1. X1

Merupakan pin yang digunakan untuk dihubungkan dengan kristal sebagai pembangkit clock.

2. X2

Berfungsi sebagai keluaran / output dari crystal yang digunakan. Terhubung juga dengan X1.

3. VBAT

Merupakan *backup supply* untuk RTC dalam menjalankan fungsi waktu dan tanggal. Besarnya adalah 3V dengan menggunakan jenis *Lithium Cell* atau sumber energi lain. Jika pin ini tidak di gunakan maka harus terhubung dengan *ground*. Sumber tegangan dengan 48mAH atau lebih besar dapat digunakan sebagai cadangan energi sampai lebih dari 10 tahun, namun dengan persyaratan untuk pengoperasian dalam suhu 25°C.

4. GND

Berfungsi sebagai *ground*.

5. SDA

Berfungsi sebagai masukan / keluaran (I/O) untuk I2C serial *interface*. Pin ini bersifat *open drain*, oleh sebab itu membutuhkan eksternal *pull up* resistor.

6. SCL

Berfungsi sebagai *clock* untuk *input* ke I2C dan digunakan untuk mensinkronisasi pergerakan data dalam serial *interface*. bersifat *open drain*, oleh sebab itu membutuhkan eksternal *pull up* resistor.

7. SWQ/OUT

Sebagai *square wave* / *Output Driver* . Jika di aktifkan, maka akan menjadi 4 frekuensi gelombang kotak yaitu 1 Hz, 4kHz, 8kHz, 32kHz sifat dari pin ini sama dengan sifat pin SDA dan SCL sehingga membutuhkan eksternal *pull up* resistor. Dapat dioperasikan dengan VCC maupun dengan VBAT.

8. VCC

Merupakan sumber tegangan utama. Jika sumber tegangan terhubung dengan baik, maka pengaksesan data dan pembacaan data dapat dilakukan dengan baik. Namun jika *backup supply* terhubung juga dengan VCC, namun besar VCC di bawah VTP, maka pengaksesan data tidak dapat dilakukan.

2.5 Sensor Tegangan

Untuk mengetahui kondisi tegangan yang terukur pada fasa R, S, dan T, maka digunakan sensor tegangan. Sensor ini terbuat dari beberapa komponen. Antara lain transformator PT, resistor, kapasitor dan dioda. Cara kerja dari sensor tegangan ini adalah dengan cara menurunkan tegangan AC menggunakan transformator PT. Keluaran tegangan AC pada kumparan sekunder dari transformator kemudian

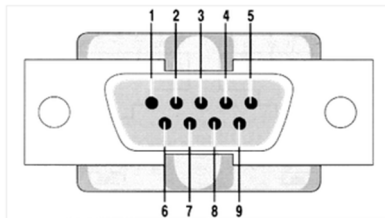
diubah menjadi tegangan DC menggunakan dioda. Hasil *output* tegangan DC inilah yang kemudian akan dimasukkan ke dalam mikrokontroler melalui metode pembacaan ADC.

2.6 RS 232 [5]

Komunikasi serial adalah salah satu fitur dari mikrokontroler AVR. Komunikasi serial biasa disebut dengan USART (*Universal Synchronous Asynchronous Receive Transmit*). Komunikasi serial ditujukan untuk mengkomunikasikan dua buah *device*, ditujukan agar mikrokontroler dapat berhubungan dengan dunia luar. Seperti mikrokontroler dengan komputer, mikrokontroler dengan GPS, mikrokontroler dengan modem, dan masih banyak *device* lainnya. Kedua *device* tersebut biasanya disebut DTE (*Data Terminal Equipment*) dan DCE (*Data Communications Equipment*).

Pada saat ini dikenal dua cara berkomunikasi serial yaitu komunikasi data serial secara *asinkron* dan komunikasi data serial secara *sinkron*. Pada komunikasi data serial *sinkron*, *clock* dikirim bersama-sama dengan data *serial*. Sedangkan komunikasi data *serial asinkron*, *clock* tidak dikirimkan bersama-sama data *serial*, tetapi dibangkitkan sendiri-sendiri.

Standar sinyal komunikasi serial yang banyak digunakan ialah standar RS 232. Standar ini hanya menyangkut komunikasi data antara komputer (*Data Terminal Equipment* – DTE) dengan alat – alat pelengkap komputer (*Data Circuit-Terminating Equipment* – DCE). Standar ini menggunakan beberapa piranti dalam implementasinya. Paling umum yang dipakai adalah konektor DB9. DB9 dapat dilihat pada gambar 2.6. Sedangkan pada tabel 2.1 dijelaskan nama dan fungsi pin DB9.



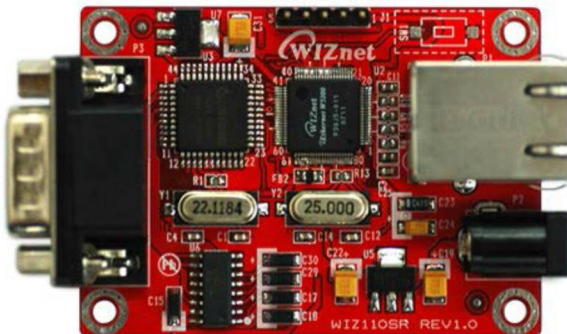
Gambar 2.6 Konektor Serial DB-9

Tabel 2.1 Nama dan Fungsi Pin DB9

<u>Nomor Pin</u>	<u>Nama Sinal</u>	<u>Fungsi</u>	<u>Keterangan</u>
1	DCD	In	Data Carrier Detect / Received Line Signal Detect
2	RxD	In	Receive Data
3	TxD	Out	Transmit Data
4	DTR	Out	Data Terminal Ready
5	GND	-	Ground
6	DSR	In	Data Set Ready
7	RST	Out	Request to Send
8	CTS	In	Clear to Send
9	R1	In	Ring Indicator

2.7 *Wiznet* [6]

Penggunaan modul *Wiznet* yaitu dari modul mikrokontroler ke laptop (PC) dengan *wireless*. Modul *Wiznet* tidak dirancang sendiri dalam Tugas Akhir ini. Berikut modul *Wiznet* WIZ110SR yang digunakan dalam Tugas Akhir ini dapat dilihat pada gambar 2.7.



Gambar 2.7 *Wiznet*

Spesifikasi dari Modul TCP/IP *Wiznet* tipe WIZ110SR dapat dilihat pada tabel 2.2.

Tabel 2.2 Spesifikasi Modul *Wiznet WIZ110SR*

<i>Items</i>	<i>Description</i>
<i>MCU</i>	8051 (having internal 62K Flash, 16K SRAM, 2K EEPROM)
<i>TCP/IP</i>	W5100 (<i>Ethernet PHY Embedded</i>)
<i>Network Interface</i>	10/100 Mbps auto-sensing RJ-45 Connector
<i>Serial Interface</i>	RS232
<i>Serial Signal</i>	TXD, RXD, RTS, CTS, GND
<i>Serial Parameters</i>	Parity : None, Even, Odd
	Data Bits : 7,8
	Flow Control : None, RTS/CTS, XON/XOFF
	Speed : up to 230Kbps
<i>Input Voltage</i>	DC 5V
<i>Power Consumption</i>	Under 180mA
<i>Temperature</i>	0°C ~ 80°C (Operation), -40°C ~ 85°C (Storage)
<i>Humidity</i>	10 ~ 90%

2.8 *Wi-Fi* [7]

Wi-fi merupakan singkatan dari *Wireless Fidelity*. *Wifi* adalah teknologi jaringan tanpa kabel yang menggunakan frekuensi tinggi. Frekuensi yang digunakan oleh teknologi *WiFi* berada pada spektrum 2,4 Ghz.

Wi-fi merupakan kependekan dari *Wireless Fidelity* yaitu sebuah media penghantar komunikasi data tanpa kabel yang bisa digunakan untuk komunikasi atau mentransfer program dan data dengan kemampuan yang sangat cepat. Istilah *wifi* banyak dikenal oleh masyarakat sebagai media untuk internet saja, namun sebenarnya bisa juga difungsikan sebagai jaringan tanpa kabel (nirkabel) seperti di perusahaan-perusahaan besar dan juga di warnet. Jaringan nirkabel tersebut biasa diistilahkan dengan LAN (*local area network*). Sehingga antara komputer dilokasi satu bisa saling berhubungan dengan komputer lain yang letaknya berbeda.



Gambar 2.8 *Wi-Fi*

Sedangkan untuk penggunaan internet, *wifi* memerlukan sebuah titik akses yang biasa disebut dengan *hotspot* untuk menghubungkan dan mengontrol antara pengguna *wifi* dengan jaringan internet pusat.

2.9 *Visual Basic* [8]

Microsoft Visual Basic menyediakan prasarana yang dapat dipergunakan secara cepat dan mudah untuk menciptakan aplikasi komputer dengan antar muka berbasis visual di lingkungan *windows*.

Visual Basic merupakan bahasa pemrograman yang sangat mudah dipelajari, dengan teknik pemrograman *Visual* yang memungkinkan penggunaannya untuk berkreasi lebih baik dalam menghasilkan suatu program aplikasi. Ini terlihat dari dasar pembuatan dalam *Visual Basic 6.0* adalah *form*, dimana pengguna dapat mengatur tampilan *form* kemudian dijalankan dalam *script* yang sangat mudah.

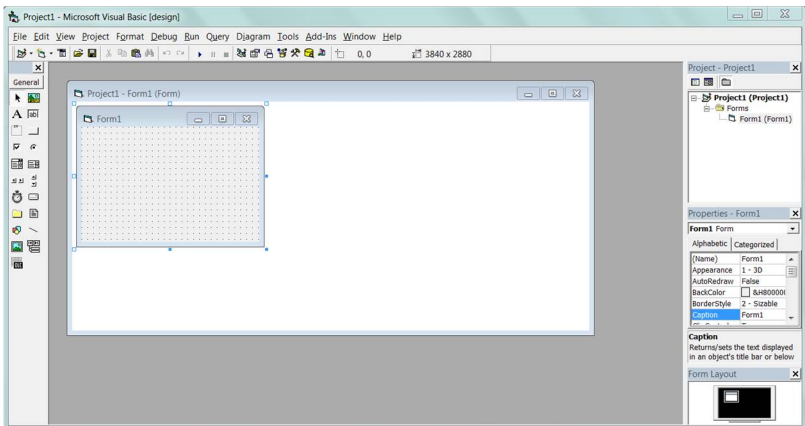
Bagian kata *VISUAL* menunjukkan program tersebut membuat aplikasi antar muka yang berbasis grafis dan bukannya dengan menuliskan baris-baris kode seperti cara pemrograman lama. Kita dapat dengan mudah menambahkan objek-objek yang telah siap ke tempat tertentu di layar.

Bagian kata BASIC mengacu pada istilah bahasa pemrograman BASIC (*Beginner All-Purpose Symbolic Instruction Code*), bahasa pemrograman gratis di lingkungan DOS (sistem operasi komputer PC sebelum era *Windows*), yang terbukti dengan kesederhanaan dan kemudahannya sejak lama.

Bahasa *Basic* pada dasarnya adalah bahasa yang mudah dimengerti sehingga pemrograman di dalam bahasa *Basic* dapat dengan mudah dilakukan meskipun oleh orang yang baru belajar membuat program.

Visual Basic dikembangkan untuk meyamai kemudahan bahasa BASIC yang lama, tetapi telah dilengkapi dengan ratusan perintah, fungsi, dan fasilitas baru, dan banyak diantaranya dapat berhubungan langsung dengan *Windows* GUI, antar muka *Windows* yang berbasis visual.

Oleh karena dibuat oleh *Microsoft*, VB 6 memiliki keunggulan dalam hal pengaksesan terhadap beberapa pustaka yang dimiliki oleh sistem operasi *Windows*. Para programmer dapat memanfaatkan *Windows* API (*Application Programming Interface*) untuk membuat program aplikasi yang lebih kompleks dan *powerfull*. Gambar 2.9 menunjukkan *layout* pertama pada *visual basic 6.0*.

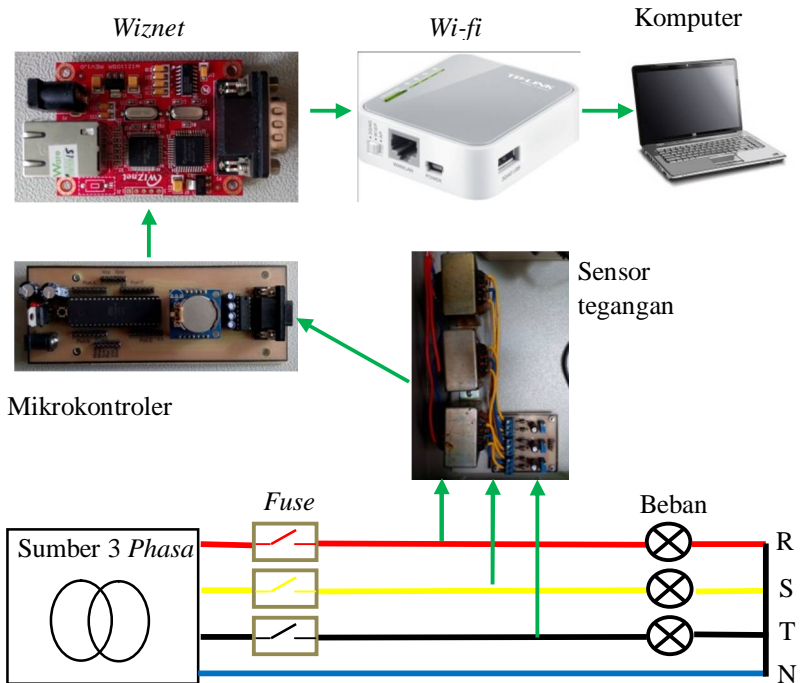


Gambar 2.9 *Layout* Pertama *Visual Basic 6.0*

Halaman ini sengaja dikosongkan

BAB III PERENCANAAN DAN PEMBUATAN

Pada bab ini akan di bahas mengenai perancangan dan pembuatan alat Data Logger Pemadaman Gardu Trafo Tiang. Perancangan dan pembuatan alat diawali dengan pemaparan diagram fungsional alat dan cara kerja lalu dilanjutkan dengan perancangan dan pembuatan *hardware* dan *software* untuk alat ini. Diagram fungsional alat dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram Fungsional Alat

Dari diagram fungsional alat seperti gambar 3.1 dapat dilihat bahwa cara kerja alat ini adalah sensor tegangan membaca nilai tegangan yang ada pada tiap fasanya yang kemudian hasilnya akan diolah oleh mikrokontroler. Setelah mikrokontroler selesai data akan dikomunikasikan dengan rangkaian serial RS232 ke *wifi*. *Wifi* akan mengirimkan data tersebut melalui protokol sesuai dengan alamat *wifi* tersebut.

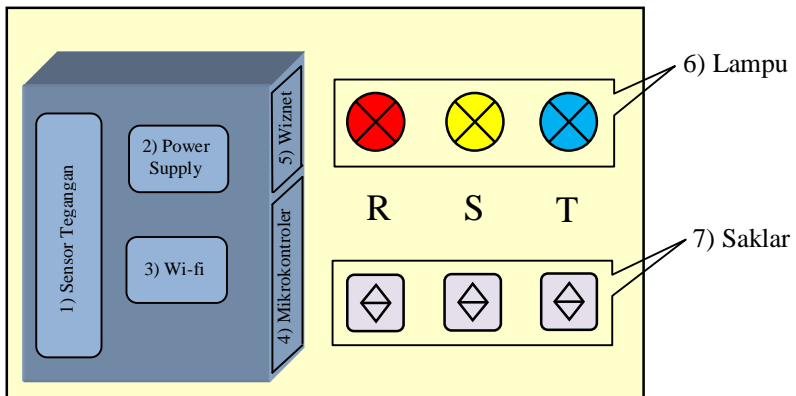
Untuk tampilan *Human Machine Interface* menggunakan *Visual Basic*. Disini *Visual Basic* akan membaca data yang dipancarkan oleh *wifi* melalui *ip address* setelah HMI dapat membaca data tersebut maka akan dapat menentukan kondisi tiap fasa pada GTT secara *real time* karena pada mikrokontroler juga terdapat RTC (*Real Time Clock*).

3.1 Perancangan dan Pembuatan *Hardware* / Perangkat Keras

Adapun perangkat keras yang dipakai dalam Tugas Akhir ini adalah rangkaian sensor tegangan, mikrokontroler, RTC, *wiznet*, serta router *TP Link*.

3.1.1 *Layout* / Tata Letak

Perancangan tata letak dimaksudkan agar penempatan peralatan menjadi rapi dan mudah dimengerti sehingga dari segi estetika dan fungsinya menjadi lebih baik terlihat pada gambar 3.2.

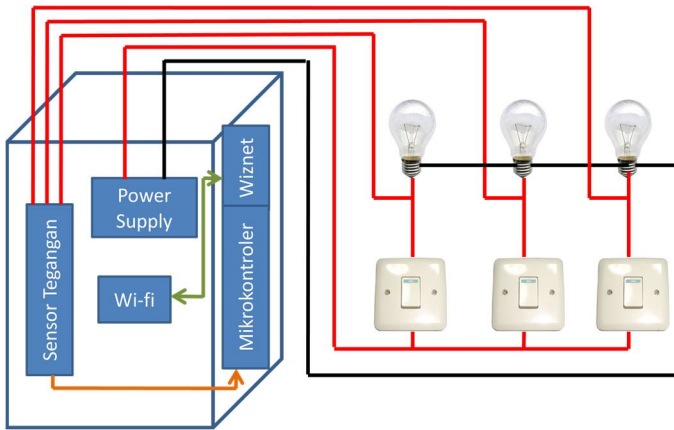


Gambar 3.2 *Layout* / Tata Letak Alat

Keterangan gambar 3.2 :

1. Sensor tegangan
2. *Power supply*
3. *Wi-fi*
4. Mikrokontroler
5. *Wiznet*
6. Lampu sebagai indikator padam
7. Saklar diibaratkan sebagai *NH fuse* dimana dapat memutuskan beban

Untuk *wiring* yang akan dilakukan pada pembuatan *hardware* dapat dilihat pada gambar 3.3.



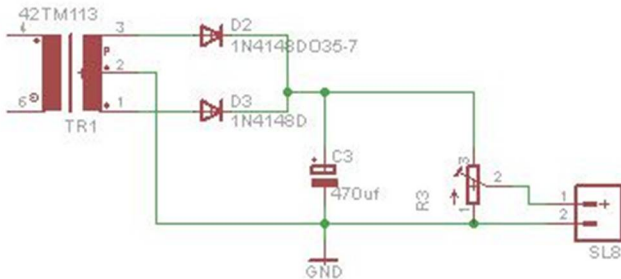
Gambar 3.3 *Wiring* pada Pembuatan *Hardware*

3.1.2 Sensor Tegangan

Sensor tegangan merupakan alat yang digunakan untuk mendeteksi besar tegangan yang melalui suatu peralatan listrik. Sensor tegangan menggunakan *transformator stepdown* untuk menurunkan nilai tegangan kerja yang digunakan.

Karena tegangan rendah yang terdapat pada *transformator stepdown* senilai 12 V dan nilai ini terlalu besar untuk dibaca pada mikrokontroler maka setelah tegangan diturunkan dengan diberi rangkaian pembagi tegangan yang terdiri dari resistor variabel dengan besar resistansi yang sama yaitu $10k\Omega$ sehingga tegangan maksimal yang tadinya 12 Volt dapat diatur menjadi 4 Volt. Tujuannya agar bisa menghasilkan tegangan dibawah 5V sesuai tegangan maksimal yang bisa di proses oleh mikrokontroler.

Untuk menyearahkan tegangan yang akan dikirim pada *PIN* ADC mikrokontroler digunakan dioda sisir 1 Ampere agar hasil penyearahan tegangan lebih sempurna. Sebelum data tegangan dikirim ke ADC *output* dari *transformator* diberi kapasitor untuk mengurangi *ripple* yang terjadi. Nilai kapasitor yang digunakan adalah $470\mu F$. Skema rangkaian sensor tegangan ditunjukkan oleh gambar 3.4. Sedangkan gambar 3.5 menunjukkan hasil pembuatan sensor tegangan.



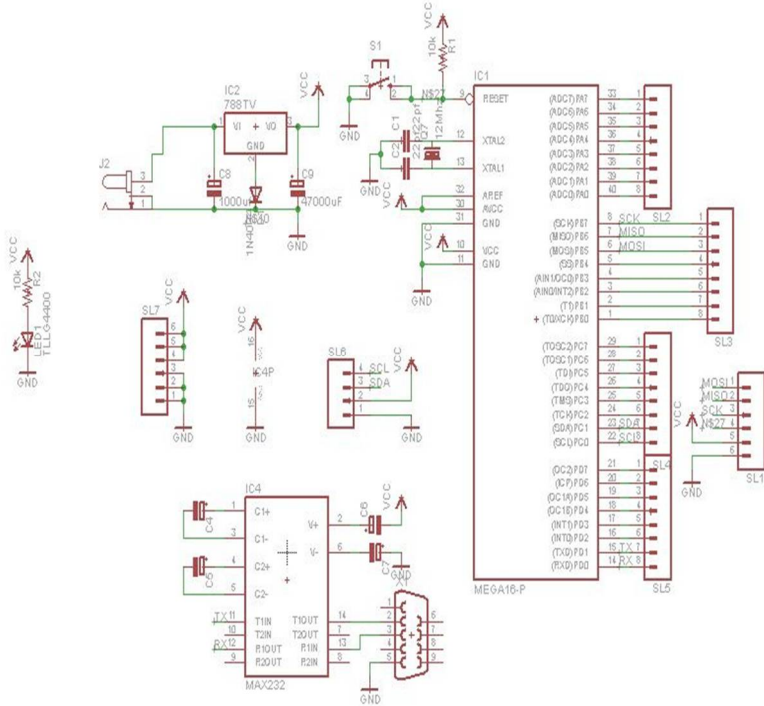
Gambar 3.4 Rangkaian Sensor Tegangan



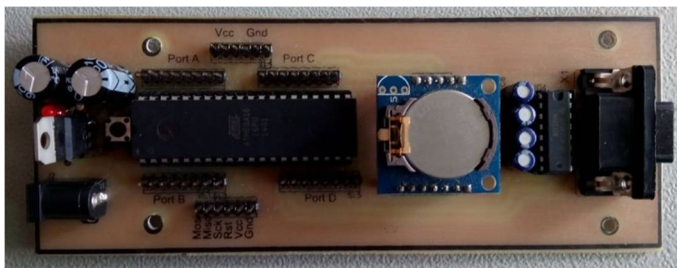
Gambar 3.5 Hasil Pembuatan Sensor Tegangan

3.1.3 Mikrokontroler

Dalam perancangan perangkat keras (*hardware*) ini terdapat rangkaian sistem minimum ATmega16. Rangkaian sistem minimum ATmega16 berfungsi untuk menerima data yang dikirimkan oleh sensor tegangan data yang diterima akan diproses untuk dikirimkan ke laptop melalui komunikasi serial 232. Rangkaian sistem minimum ATmega16 dapat dilihat pada gambar 3.6. Dan hasil pembuatan mikrokontroler ditunjukkan pada gambar 3.7.



Gambar 3.6 Rangkaian Sistem Minimum



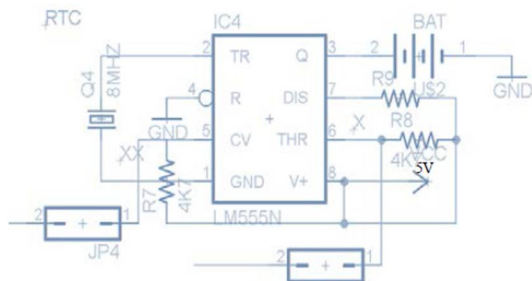
Gambar 3.7 Hasil Pembuatan Mikrokontroler

3.1.4 RTC

RTC yang digunakan dalam proyek ini adalah RTC DS1307 dengan antar muka I2C. RTC DS1307 menyediakan pewaktu dalam detik, menit, jam, hari, tanggal, bulan dan tahun. Selain itu, RTC ini menyediakan pin *battery-backup* untuk dihubungkan pada baterai

lithium 3Volt atau sumber energi lain sehingga ketika *supply* energi utama (VCC dan GND) mati, *battery-backup* mengambil alih *supply* energi pada RTC dan *timer* tetap berjalan sebagaimana mestinya.

Penggunaan 3 Volt *lithium* 48mAh *battery-backup*, RTC hanya mengkonsumsi arus kurang dari 500nA sehingga dengan baterai tersebut mampu bertahan hingga 11 tahun. Pada rangkaian RTC dengan IC DS1307, Pin SCL dihubungkan dengan *Port* C.0 mikrokontroler dan Pin SDA dihubungkan dengan *Port* C.1 mikrokontroler. Rangkaian RTC dapat dilihat pada gambar 3.8.

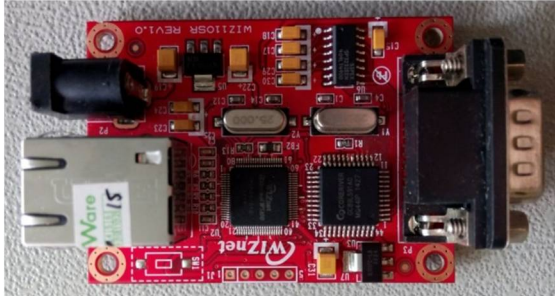


Gambar 3.8 Rangkaian RTC dengan IC DS1307

3.1.5 *Wiznet*

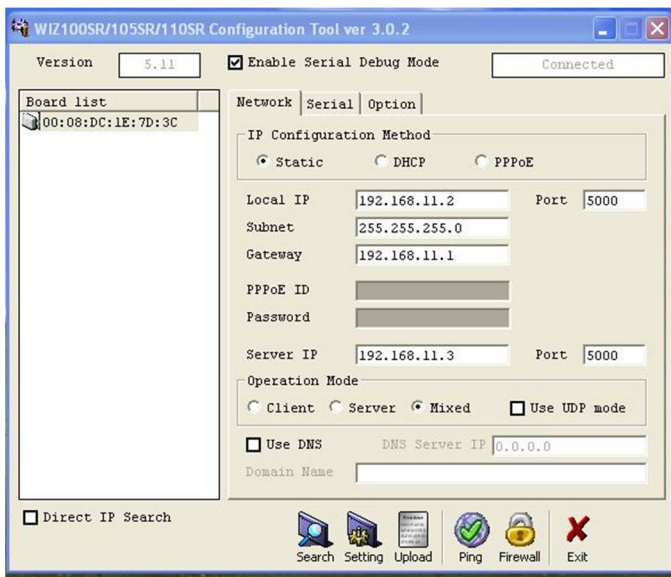
Modul *Wiznet* ini dapat di-*setting* menggunakan aplikasi WIZ110SR *Configuration tool*. Dengan menghubungkan modul *Wiznet* dengan kabel Ethernet kemudian dihubungkan ke laptop maka kita dapat melakukan *setting* IP untuk *Wiznet* yang akan dihubungkan dengan *router*.

Setting IP *wiznet* dengan *router* harus sesuai apabila tidak sesuai maka *wiznet* tidak akan bisa terhubung. Modul *wiznet* ini membutuhkan *supply* 5 V agar dapat bekerja. Bekerjanya *wiznet* ini ditandai dengan adanya lampu hijau dan *orange* pada tempat kabel ethernet. Apabila sudah ada lampu tersebut maka *wiznet* sudah dapat digunakan. Modul *wiznet* tidak dirancang sendiri. Berikut modul *wiznet* dilihat pada gambar 3.9.



Gambar 3.9 Modul Wiznet

WIZ110SR adalah *converter* protokol yang mentransmisikan data yang dikirim oleh computer melalui *port* serial sebagai data TCP / IP dan mengkonversi kembali data TCP / IP yang diterima melalui jaringan menjadi data serial kepada komputer kemudian mengirimkan kembali ke mikrokontroler. Dalam penggunaan modul TCP/IP, diperlukan suatu perangkat lunak (*software*) untuk dapat mengkonfigurasi melalui media. Berikut adalah tampilan *Configuration Tool* WIZ110SR pada gambar 3.10.



Gambar 3.10 *Configuration Tool* WIZ110SR

Dalam menggunakan modul TCP/IP terdapat beberapa langkah yang harus dilakukan, diantaranya.

- a. Mempersiapkan modul TCP/IP, rangkaian RS232, kabel LAN dan kabel penghubung RS232 ke mikrokontroler. Setelah semua alat terhubung, selanjutnya mengaktifkan *firmware* untuk modul yang berupa *software WIZ110SR configuration tool* pada komputer kemudian *setting* modul.
- b. Setelah memberikan alamat pada modul, selanjutnya mengatur alamatIP pada komputer. Alamat ini harus sama dengan alamat modul, terutama pada penggunaan mode *Static*.

Local IP : 192.168.11.2

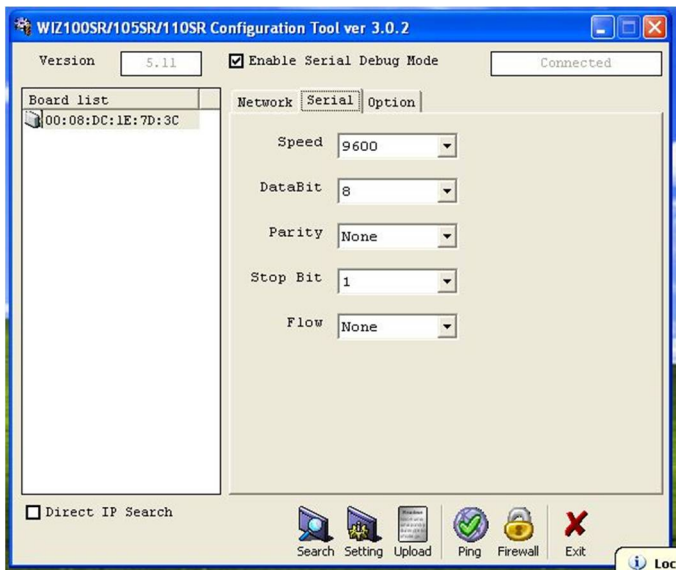
Subnet : 255.255.255.0

Gateway : 192.168.11.1

Server IP : 192.168.11.3

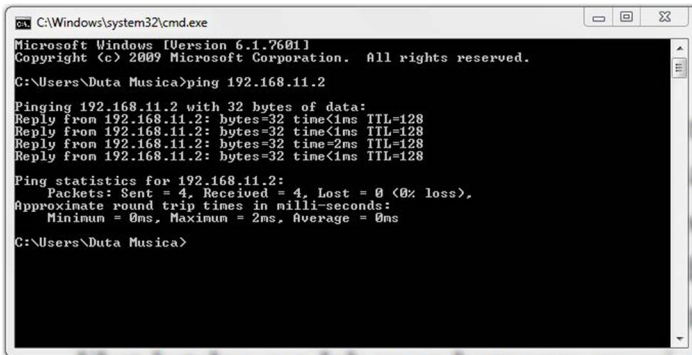
Sedangkan *port* pada *Local IP* dan *Server IP* diisi : 5000

Sedangkan pada tab Serial setting yang diisikan ditunjukkan oleh gambar 3.11 berikut :



Gambar 3.11 Tampilan Jendela Serial pada *Configuration Tool* WIZ110SR

- c. Setelah proses *setting* modul dan komputer selesai, selanjutnya dilakukan tes koneksi. Jika semua *setting* sudah benar, maka modul dapat digunakan.
- d. Untuk mengecek apakah komputer sudah terkoneksi dengan Wiznet yaitu dengan cara *ping* IP Wiznet pada *Commant Prompt*. gambar 3.12 dibawah ini menunjukkan bahwa komputer telah terkoneksi dengan *wiznet*.
- e.



Gambar 3.12 Pengecekan Koneksi pada *Commant Prompt*

3.1.6 Router TP Link

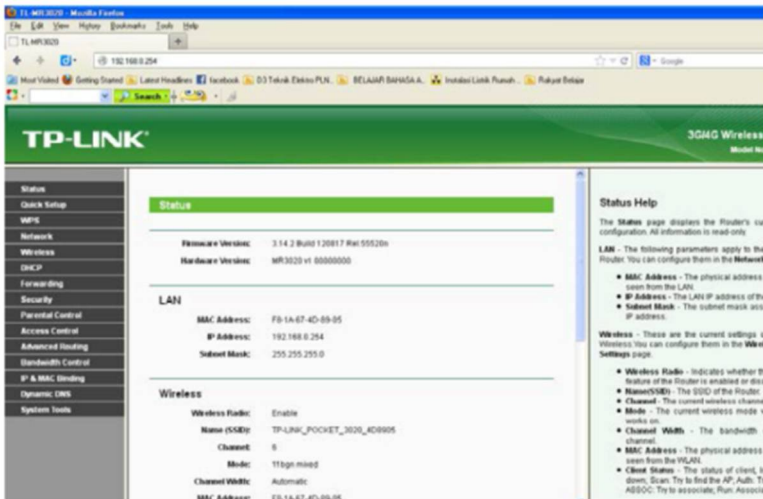
Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan yang lainnya. Dalam koneksi tugas akhir ini mempergunakan *router* TP-LINK TL-MR3020 sebagaimana ditunjukkan pada gambar 3.13.



Gambar 3.13 Router TP-LINK TL-MR3020

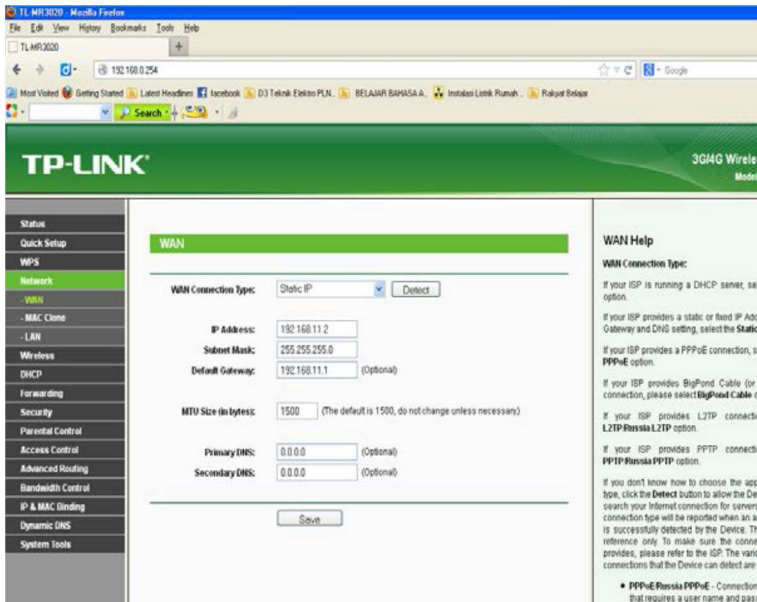
Untuk memulai koneksi pada *router*, hal yang dilakukan adalah *menset* IP pada *Router* .Adapun langkah-langkahnya sebagai berikut :

1. Aktifkan *Router* pada mode WISP. Mode ini adalah salah satu mode yang tersedia pada *Router* TP-LINK TL-MR3020.
 2. Sambungkan *Router* dengan kabel RJ-45 ke komputer.
 3. Buka *browser* internet , misalkan Mozilla Firefox.
 4. Masukkan alamat <http://192.168.0.254/> , masukkan *User name* : Admin
Password : Admin
- Maka akan tampil seperti gambar 3.14 berikut :



Gambar 3.14 Tampilan *Browser* Mozilla Firefox

5. Klik *Network*, pilih WAN kemudian *set static* IP , seperti pada gambar 3.15 berikut :



Gambar 3.15 Tampilan *Network* pada Aplikasi TP-LINK

6. Setelah proses *setting* modul dan komputer selesai, selanjutnya dilakukan tes koneksi. Jika semua *setting* sudah benar, maka modul dapat digunakan.

3.2 Perancangan dan Pembuatan *Software/ Perangkat Lunak*

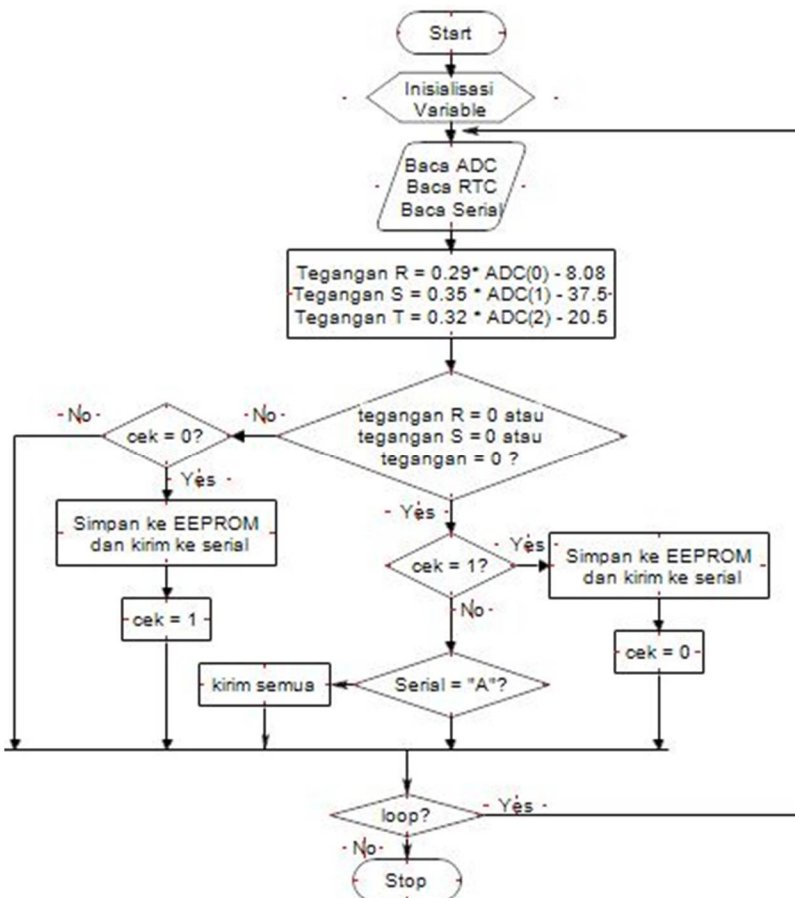
Perangkat lunak yang digunakan pada Tugas Akhir ini berupa pemrograman mikrokontroler Atmega16 menggunakan *Codevision AVR* dan tampilan di sisi *server* menggunakan *Visual Basic 6.0*.

3.2.1 *Codevision*

CodevisionAVR merupakan *software* pemrograman berbasis Bahasa C. *CodevisionAVR* ini di khususkan untuk para *programmer* di bidang elektronika, seperti program mikrokontroler untuk membuat *hardware – hardware* seperti jam digital atau sejenisnya. Dalam tugas akhir ini digunakan pemograman *CodevisionAVR* untuk membaca nilai tegangan yang dikeluarkan oleh sensor tegangan. Pemrograman dilakukan dengan memanfaatkan *port* ADC yang ada pada mikrokontroler.

Input yang berasal dari sensor tegangan masuk pada mikrokontroler kemudian mikrokontroler mengolah data ADC yang

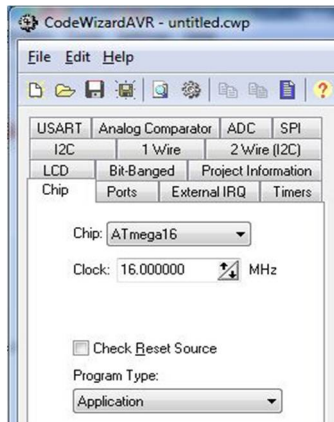
berupa nilai tegangan *input* dan dimasukkan dalam rumus sesuai dengan data yang telah diambil sebelumnya, jika nilai dari perhitungan memenuhi syarat tertentu maka nilai tegangan, tanggal dan jam disimpan dalam *eprom* dan simpanan tersebut dikirim ke komputer via *wireless*, dan jika nilai tidak memenuhi syarat tertentu maka mikrokontroler tidak akan menyimpan ke *eprom* maupun mengirim via *wireless*. Pemrograman ini akan berlangsung terus menerus jadi ADC akan selalu membaca tegangan secara *real time* dengan syarat *supply* untuk mikrokontroler dalam kondisi aktif. Berikut ini adalah *flowchart* pemrograman pada mikrokontroler dapat dilihat pada gambar 3.16.



Gambar 3.16 Flowchart Mikrokontroler

a. *Setting Chip ATmega16*

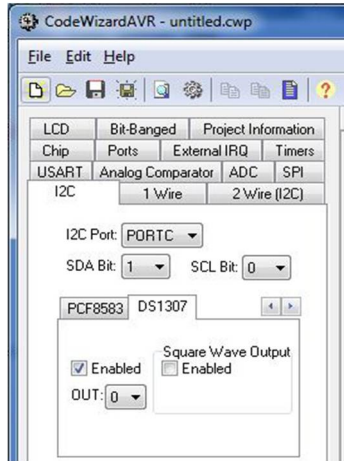
Dalam pemrograman *CodevisionAVR* harus di-*setting* terlebih dahulu sistem minimum menggunakan chip mikrokontroler dan *clock* yang digunakan. Disini menggunakan chip *ATMega16* dan *clock* 16 MHz sesuai *clock* yang dimiliki sistem minimum AVR. *Setting Chip ATMega16* dapat dilihat pada gambar 3.17.



Gambar 3.17 *Setting Chip ATMega16*

b. *Setting I2C*

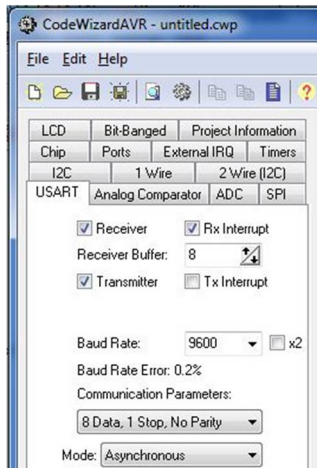
Dalam port I2C digunakan untuk mengakses *Real Time Clock* (RTC) dan untuk mengakses *eprom* eksternal 24C32. DS1307 merupakan IC *Real Time Clock* (RTC) yang dapat diakses dengan mikrokontroler menggunakan komunikasi serial I2C. Dengan adanya RTC ini maka dapat menampilkan waktu yang berupa jam, menit, dan detik, serta tanggal, yaitu hari, bulan dan tahun. *Setting* RTC diletakkan di *port C* dalam sistem minimum. AT24C32 merupakan IC *eprom* eksternal yang dapat diakses melalui mikrokontroler menggunakan komunikasi serial I2C. Antara *eprom* eksternal dan RTC terdapat masing-masing alamat, sehingga tidak akan terjadi benturan data antara RTC dan *eprom* eksternal. gambar 3.18.



Gambar 3.18 Setting I2C

c. *Setting Pengiriman Data*

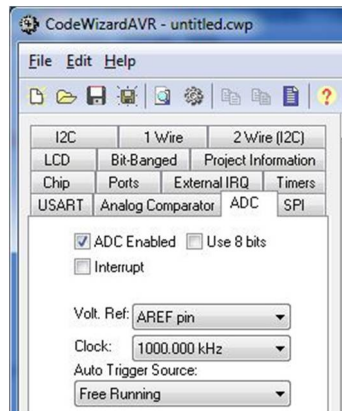
Karena dalam Tugas Akhir menggunakan sistem pengiriman media *wifi*, jadi dalam *codevision* harus di-*setting* program pengiriman (*transmitter*) dan penerimaan (*receiver*) di dalam mikrokontroler agar dapat mengirim data yang terbaca oleh ADC dan menerima perintah dari komputer. *Setting* pengiriman data dapat dilihat pada gambar 3.19.



Gambar 3.19 Setting Pengiriman Data

d. *Setting Input ADC*

ADC adalah suatu rangkaian yang mengubah data berupa tegangan analog ke data digital. ADC ini digunakan bila ada *input* tegangan analog. Hal – hal yang juga perlu diperhatikan dalam penggunaan ADC ini adalah tegangan maksimum yang dapat dikonversikan oleh ADC dari rangkaian pengkondisi sinyal tipe keluaran. *Setting* ADC menggunakan *port A* dengan data 10 bit, dan semua *port A.0* sampai *port A.7* dapat digunakan sebagai *port ADC*. *Setting Input* ADC dapat dilihat pada gambar 3.20.



Gambar 3.20 *Setting Input* ADC

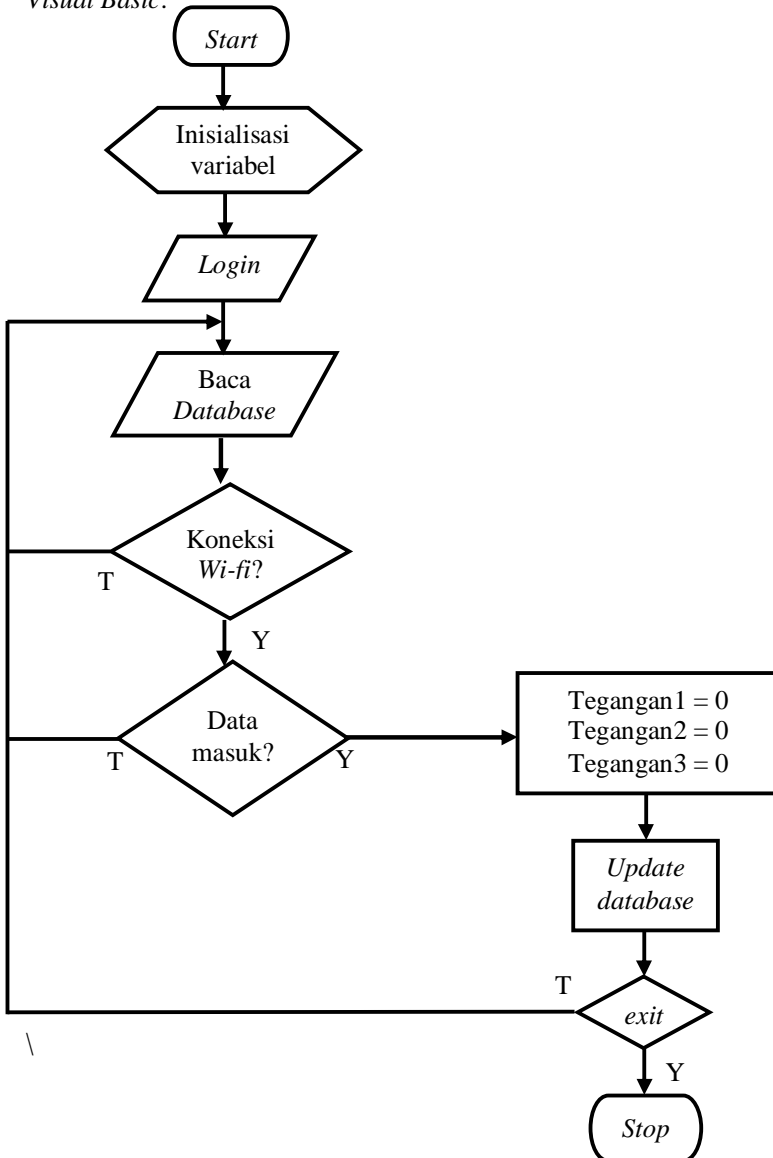
3.2.2 *Visual Basic*

Perancangan VB digunakan untuk monitoring oleh pihak operator telepon (optel) pelayanan teknik dari layar monitor komputer (PC). *Software* yang digunakan adalah *Visual Basic 6.0* . Bentuk tampilannya adalah berupa kolom fasa dan waktu padam.

Dari *Flowchart* tampilan algoritma dari perancangan perangkat lunak untuk *Visual Basic* ini adalah :

1. Saat aplikasi diakses, akan muncul *Form Login* karena aplikasi cuma bisa diakses orang tertentu saja
2. Setelah *Login* ditampilkan bentuk tabel *data logger*
3. Untuk mengakses data harus konek ke *Wifi* dulu sehingga akan muncul data pemadaman pada gardu trafo tiang.
4. Untuk meng-*download* data yang tersimpan di *eprom* bisa mengklik tombol *download* dan untuk keluar dari tampilan monitoring harus mengklik tombol *exit*.

Pada gambar 3.21 akan ditampilkan *Flowchart* dari program *Visual Basic*:



Gambar 3.21 *Flowchart Visual Basic*

BAB IV PENGUJIAN DAN PENGUKURAN

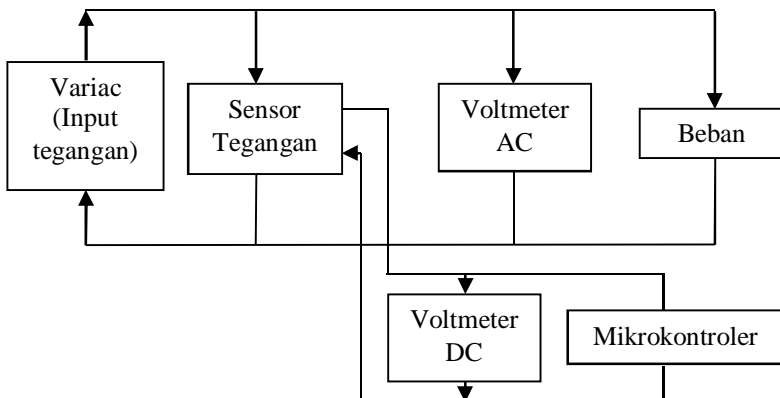
Dalam bab ini akan dibahas tentang pengujian berdasarkan perencanaan dari sistem yang dibuat. Pengujian ini dilaksanakan untuk mengetahui apakah sistem telah sesuai dengan yang diharapkan atau belum. Pengujian pertama dilakukan secara terpisah, dan kemudian dilakukan ke dalam sistem yang telah terintegrasi.

Pengujian yang dilakukan pada bab ini antara lain:

1. Pengujian Sensor Tegangan
2. Pengujian *EEPROM*
3. Pengujian Komunikasi Serial
4. Pengujian *Router Wi-fi*
5. Pengujian *Database*
6. Pengujian *Visual basic*
7. Pengujian Penghitung lama waktu padam
8. Pengujian alat secara keseluruhan

4.1 Pengujian Sensor Tegangan

Pada tahap pengujian sensor tegangan menggunakan perubahan tegangan dari variac yang *inputan* variac dari tegangan PLN dan *outputnya* dapat diubah – ubah. Diagram pengujian sensor dapat dilihat pada gambar 4.1.



Gambar 4.1 Diagram Pengujian Sensor Tegangan

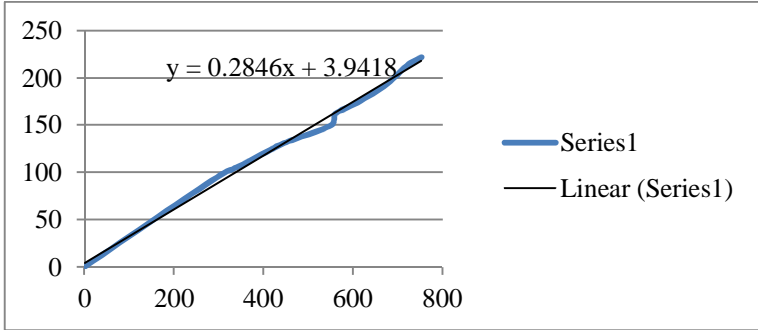


Gambar 4.2 Pengujian Sensor Tegangan

Dalam pengujian sensor, diambil 5 data dengan berbagai perubahan tegangan *output* variac. Hasil pengujian sensor dapat dilihat pada tabel 4.1, 4.2, dan 4.3. sedangkan gambar grafik linearisasi dapat dilihat pada gambar 4.3, 4.4, dan 4.5.

Tabel 4.1 Hasil Pengujian Sensor Tegangan Fasa R

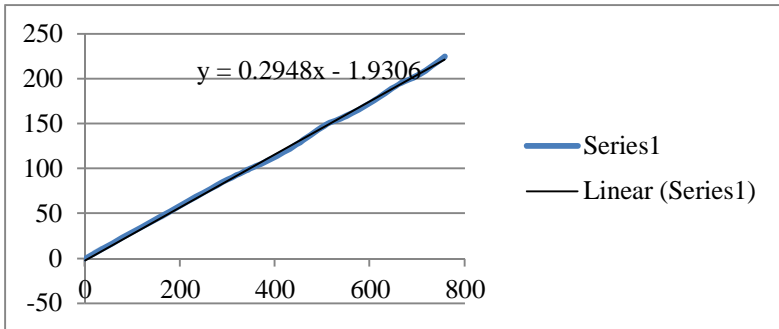
Beban (Watt)	V Input AC (Volt)	V Output DC (Volt)	ADC
100	0	0	0
100	95	1.96	296
100	106	2.16	343
100	119	2.4	395
100	131	2.61	446
100	150	2.86	551
100	162	3.08	560
100	174	3.3	610
100	194	3.67	677
100	213	3.92	720
100	222	4.1	753



Gambar 4.3 Grafik Linearisasi Fasa R

Tabel 4.2 Hasil Pengujian Sensor Tegangan Fasa S

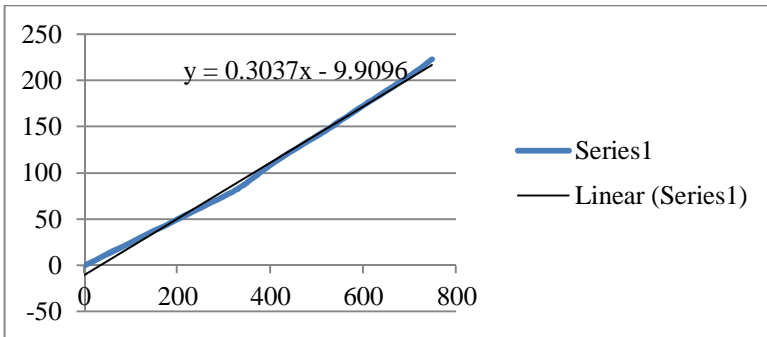
Beban (watt)	V Input AC (Volt)	V Output DC (Volt)	ADC
100	0	0	0
100	85	1.8	290
100	100	2.1	348
100	111	2.28	392
100	127	2.43	446
100	148	2.81	504
100	159	2.99	551
100	174	3.27	603
100	194	3.63	660
100	205	3.8	704
100	225	4.1	758



Gambar 4.4 Grafik Linearisasi Fasa S

Tabel 4.3 Hasil Pengujian Sensor Tegangan Fasa T

Beban (watt)	V Input AC (Volt)	V Output DC (Volt)	ADC
100	0	0	0
100	78	1.64	312
100	89	1.84	348
100	107	2.11	395
100	127	2.54	457
100	146	2.84	519
100	158	3	555
100	174	3.24	603
100	194	3.65	665
100	205	3.8	700
100	223	4.08	748



Gambar 4.5 Grafik Linearisasi Fasa S

Dari data pengujian dan grafik pengukuran, dapat dihasilkan suatu persamaan karakteristik sensor tegangan yang akan digunakan dalam pembacaan mikrokontroler agar didapatkan tegangan yang sesuai dengan tegangan pengukuran pada voltmeter. Berikut persamaan linier yang digunakan untuk mendapatkan pembacaan mikrokontroler.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \dots \dots \dots (4.1)$$

Keterangan:

- y = Nilai ADC
- x = Nilai tegangan input (VAC)

- y_1 = Tegangan *output* data ke-1
- y_2 = Tegangan *output* data ke-n
- x_1 = Tegangan *input* data ke-1
- x_2 = Tegangan *input* data ke-n

Dari tabel pengujian diambil sebelas data yaitu dari tegangan 0V sampai dengan 225V yang digunakan untuk pembuatan persamaan karakteristik sensor tegangan, sehingga didapat persamaan :

fasa R : $y = 0.2846x + 3.9418$ (4.2)

fasa S : $y = 0.2948x - 1.9306$ (4.3)

fasa T: $y = 0.3037x - 9.9096$ (4.4)

Dari persamaan yang didapatkan, dilakukan pengujian untuk mendapatkan *error* pengukuran. Pengukuran dilakukan dengan cara memberikan tegangan yang berbeda – beda dari variac, kemudian dengan menggunakan avometer, dilakukan pengukuran tegangan. Sehingga didapatkan nilai tegangan dari avometer dan dari mikrokontroler. Pada saat pengujian ini beban tidak diubah-ubah. Perhitungan *error* diperoleh dari persamaan berikut :

$$\% \text{ Error} = \frac{\text{Hasil multimeter} - \text{Hasil mikrokontroler}}{\text{Hasil multimeter}} \times 100\%$$

Sehingga didapatkan *error* tiap fasanya seperti tabel 4.4, 4.5, dan 4.6:

Tabel 4.4 Hasil Pengujian *Error* Sensor Tegangan Fasa R

V input dari variac (Volt)	Hasil pengukuran mikrokontroler di LCD (Volt)	<i>Error</i> (%)
78	79	1.3
86	87	1.2
100	102	2.0
115	115	0.0
127	127	0.0
139	138	0.7
145	143	1.4

Lanjutan Tabel 4.4

V input dari variac (Volt)	Hasil pengukuran mikrokontroler di LCD	Error (%)
165	169	2.4
189	187	1.1
222	225	1.4

Tabel 4.5 Hasil Pengujian *Error* Sensor Tegangan Fasa S

V input dari variac (Volt)	Hasil pengukuran mikrokontroler di LCD (Volt)	<i>Error</i> (%)
84	85	1.2
103	105	1.9
123	124	0.8
143	145	1.4
157	154	1.9
169	169	0
176	176	0
192	193	0.5
212	215	1.4
225	224	0.4

Tabel 4.6 Hasil Pengujian *Error* Sensor Tegangan Fasa T

V input dari variac (Volt)	Hasil pengukuran mikrokontroler di LCD (Volt)	<i>Error</i> (%)
81	82	1.2
98	96	2.0
109	109	0.0
116	113	2.6
127	130	2.4
139	138	0.7
158	158	0.0
163	164	0.6
193	193	0.0
221	224	1.4

Dari hasil pengukuran tegangan menggunakan sensor tegangan di setiap fasa maka dapat diketahui *error* sensor tegangan. Untuk *error*

rata-rata pada data yang telah didapatkan maka dapat dihitung dengan rumus sebagai berikut :

$$e_r \text{ rata - rata} = \frac{\text{jumlah error}}{\text{banyaknya data yang diambil}}$$

Dari rumus tersebut dapat diketahui *error* rata-rata sensor tegangan pada setiap fasanya seperti tabel 4.7:

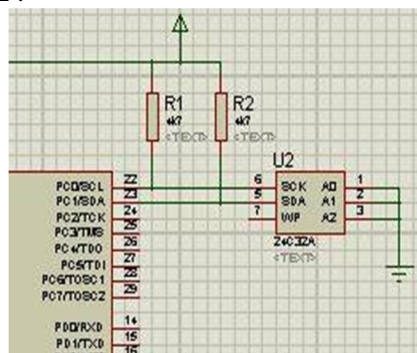
Tabel 4.7 Rata-rata *Error* Sensor Tegangan

<i>Error</i> rata rata (%)		
Fasa R	Fasa S	Fasa T
1,15 %	0,95%	1.09%

Dari hasil pengujian sensor tegangan yang telah dilakukan, dapat disimpulkan pemakaian sensor ini memberikan *error* pengukuran sebesar 1,15% pada fasa R, 0.95% pada fasa S, 1.09% pada fasa T.

4.2 Pengujian *EEPROM*

Pada tugas akhir ini, menggunakan *EEPROM* (*Electrically Erasable Programmable Read-Only Memory*) dengan seri AT24C32 dengan kapasitas sebesar 32,768 bits atau sekitar 4 Kb (Kilo Byte) *EEPROM* AT24C32 merupakan *EEPROM* eksternal yang dapat diakses melalui I2C pada mikrokontroler. Berikut gambar 4.6 skematik *EEPROM* AT24C32 :



Gambar 4.6 Skematik *EEPROM* AT24C32

Pengujian *EEPROM* dilakukan dengan menggunakan 3 *push button* untuk menyimpan data. Yang mana bila salah satu *push button*

ditekan, maka data akan tersimpan dan 1 detik kemudian data yang tersimpan tersebut dipanggil kembali dan dikirim lewat serial DB9. Komputer akan memonitor data yang dikirimkan mikrokontroler Atmega16 menggunakan program *Putty* dengan *baudrate* 9600. Agar mikrokontroler dapat digunakan untuk mengakses *EEPROM* AT24C32, maka mikrokontroler diprogram menggunakan *CodevisionAVR*. Program *CodevisionAVR* ditunjukkan pada gambar 4.7 :

```
while (1)
{
    // Place your code here
    if(PIND.0==0)
    {
        printf("Write byte to EEPROM memory...");
        eeprom_write(0, 0xAA);
        delay_ms(1000);
        printf("Read byte from EEPROM memory...");
        putchar(13);
        printf("Read byte = 0x");
        printf("%i", eeprom_read(0));
        delay_ms(10);
    }else

    if(PIND.1==0)
    {
        printf("Write byte to EEPROM memory...");
        eeprom_write(0, 125);
        delay_ms(1000);
        printf("Read byte from EEPROM memory...");
        putchar(13);
        printf("Read byte = ");
        printf("%i", eeprom_read(0));
        delay_ms(10);
    }

    if(PIND.2==0)
    {
        printf("Write byte to EEPROM memory...");
        eeprom_write(0, 12);
        delay_ms(1000);
        printf("Read byte from EEPROM memory...");
        putchar(13);
        printf("Read byte = ");
        printf("%i", eeprom_read(0));
        delay_ms(10);
    }
}
```

Gambar 4.7 Program *CodevisionAVR* untuk Pengujian *EEPROM*

Hasil pengujian untuk penyimpanan *EEPROM* dapat dilihat pada gambar 4.8 dan tabel 4.8:

```

COM9 - PuTTY
Write byte to EEPROM memory...
Read byte from EEPROM memory...
Read byte = 0xAA
Write byte to EEPROM memory...
Read byte from EEPROM memory...
Read byte = 0xAA
Write byte to EEPROM memory...
Read byte from EEPROM memory...
Read byte = 125
Write byte to EEPROM memory...
Read byte from EEPROM memory...
Read byte = 12

```

Gambar 4.8 Hasil Pengujian *EEPROM*

Tabel 4.8 Hasil Pengujian *EEPROM*

Data yang disimpan	Data yang tersimpan
AA	AA
125	125
12	12
223	223
1023	255

Dari hasil pengujian pada tabel 4.8 diatas, dapat disimpulkan bahwa *EEPROM* menyimpan data tiap-tiap 8 bit. Sehingga bila data yang disimpan lebih dari 8 bit, maka akan tersimpan pada alamat selanjutnya.

4.3 Pengujian Komunikasi Serial

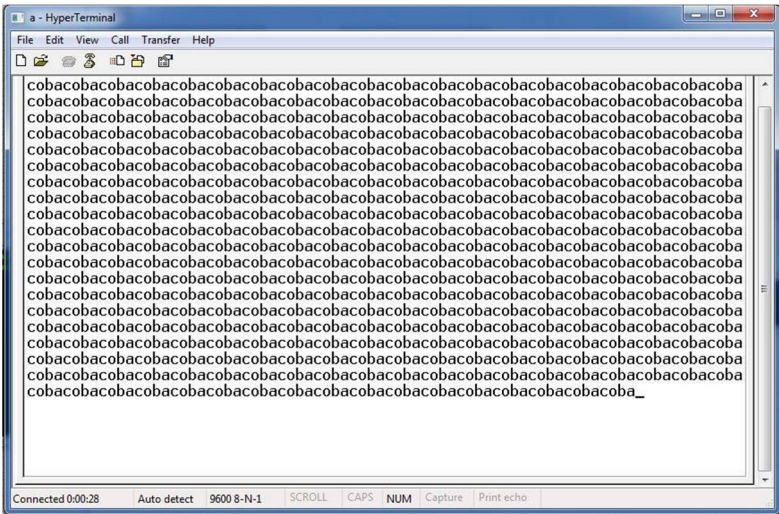
Rangkaian ini digunakan untuk komunikasi serial antara mikrokontroler dan PC dengan media *Wifi*. Oleh karena itu, diperlukan sebuah modul TCP/IP (*Wiznet*) untuk menghubungkan antara mikrokontroler dengan *router Wifi*. Pengujian rangkaian dilakukan dengan menghubungkan mikrokontroler Atmega16 dengan modul *Wiznet* melalui konektor DB-9 kemudian dihubungkan ke PC menggunakan kabel UTP.

Selanjutnya membuat program pada *CodeVisionAVR* dan melihat tampilan datanya pada *Hyperterminal*. Program yang diberikan pada

mikrokontroler seperti gambar 4.9 dan hasilnya dapat dilihat pada *Hyperterminal* seperti gambar 4.10.

```
while (1)
{
// Place your code here
printf("coba");
delay_ms(100);
};
}
```

Gambar 4.9 Program *Code Vision AVR* untuk Pengujian Komunikasi Serial



Gambar 4.10 Hasil Tampilan Pengujian Komunikasi Serial

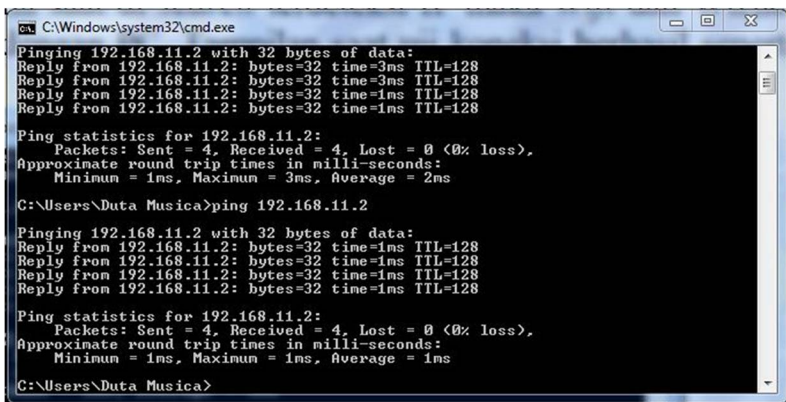
4.4 Pengujian Router Wi-fi

Untuk mengetahui *Router* TP-LINK TL MR3020 dapat digunakan perlu dilakukan suatu pengujian koneksi. *Router* pada tugas akhir ini difungsikan sebagai AP (*Access Point*) Mode. Pada mode ini, *router* akan bertindak sebagai pusat penghubung klien LAN nirkabel, dan dapat memberikan perluasan jaringan nirkabel untuk jaringan kabel. Hasil pengujian koneksi antara komputer dengan *router* TP-LINK dapat dilihat pada gambar 4.11.



Gambar 4.11 Pengujian Koneksi Komputer dengan Router

Setelah terkoneksi dengan *WiFi* TP-LINK perlu diuji apakah *WiFi* sudah benar-benar terkoneksi atau belum. Uji koneksi dapat menggunakan program komunikasi juga dapat dilakukan dengan menggunakan program bawaan *Microsoft* yaitu *command prompt*. Pengujian ini disebut juga sebagai *Ping Test*. *Ping Test* dilakukan dengan cara membuka kotak dialog *run* dan mengetikkan perintah *cmd*. Setelah itu akan muncul kotak dialog dan kita ketikkan perintah *Ping* 192.168.11.2 (IP *WiFi* yang sudah di-*setting* sebelumnya). Apabila koneksi berhasil, *WiFi* akan memberikan respon *reply*. Hasil *Ping Test* lewat *cmd* bisa dilihat pada gambar 4.12.



Gambar 4.12 Hasil *Ping Test* Pengujian Router

4.5 Pengujian Database

Seluruh proses akuisisi data akan masuk ke dalam sebuah *database*. *Database* ini disimpan menggunakan *Ms. Access*. Pengujian *database* dilakukan untuk memastikan bahwa penyimpanan data ke dalam *database* tersebut berjalan dengan baik. Untuk menyimpan seluruh proses tersebut, HMI sudah di setting otomatis menyimpan setiap data masuk sehingga *database* dapat tersimpan dalam *Ms. Access*. Dalam *database* ini dapat dilihat rekaman hasil monitoring dengan melihat tanggal dan waktu pengambilan data padam dan durasi padam. Tampilan *database* dalam *Ms. Access* dapat dilihat pada gambar 4.13.

	Jam	Tanggal	Keterangan	Durasi R	Durasi S	Durasi T
1	12:17:07 AM	4/6/2015	R Off, S On, T On			
2	12:17:37 AM	4/6/2015	R On, S On, T On	00:00:30		
3	12:17:38 AM	4/6/2015	R On, S Off, T On			
4	12:18:11 AM	4/6/2015	R On, S On, T On		00:00:33	
5	12:18:13 AM	4/6/2015	R On, S On, T Off			
6	12:18:44 AM	4/6/2015	R On, S On, T On			00:00:31
7	12:18:55 AM	4/6/2015	R Off, S Off, T On			
8	12:18:55 AM	4/6/2015	R Off, S Off, T On			
9	12:19:55 AM	4/6/2015	R On, S On, T On	00:01:00	00:01:00	
10	12:19:55 AM	4/6/2015	R On, S On, T On			
11	12:20:04 AM	4/6/2015	R Off, S On, T Off			
12	12:20:04 AM	4/6/2015	R Off, S On, T Off			
13	12:21:21 AM	4/6/2015	R On, S On, T On	00:01:17		00:01:17
14	12:21:21 AM	4/6/2015	R On, S On, T On			
15	12:21:23 AM	4/6/2015	R On, S Off, T Off			
16	12:21:23 AM	4/6/2015	R On, S Off, T Off			
17	12:21:58 AM	4/6/2015	R On, S On, T On		00:00:35	00:00:35
18	12:21:58 AM	4/6/2015	R On, S On, T On			
19	12:22:03 AM	4/6/2015	R Off, S Off, T Off			
20	12:22:03 AM	4/6/2015	R Off, S Off, T Off			
21	12:22:05 AM	4/6/2015	R Off, S Off, T Off			
22	12:22:56 AM	4/6/2015	R Off, S On, T On		00:00:53	00:00:53
23	12:22:56 AM	4/6/2015	R Off, S On, T On			
24	12:22:58 AM	4/6/2015	R On, S On, T On	00:00:55		
*						

Gambar 4.13 Tampilan Database

4.6 Pengujian *Visual Basic*

Pada tahap ini, untuk mengetahui data logger pemadaman gardu trafo tiang menggunakan program *Visual Basic*. Pengujian untuk visual basic ini dibedakan menjadi dua yaitu pengujian *security*/keamanan serta tampilan *data logger*.

4.6.1 Pengujian *Security*

Security atau sistem keamanan merupakan salah satu hal penting. Hal ini diperlukan untuk mencegah jika ada seseorang yang tidak bertanggung jawab menyalah gunakan sistem data logger tersebut.

Untuk itulah dibutuhkan sebuah *security* seperti memberi halaman *login* pada HMI yang ditampilkan. Pemberian *username* dan *password* sebagai proteksi terhadap pengakses yang tidak memiliki ijin. Pada gambar 4.14 terlihat tampilan *form login* awal sebelum masuk ke HMI data logger pemadaman gardu trafo tiang. Untuk dapat masuk mengakses harus memasukkan Id Pengguna dan *password* yang sesuai. Jika pengakses tidak memiliki kewenangan untuk melakukan monitoring, maka pengakses tidak dapat melanjutkan langkah selanjutnya.

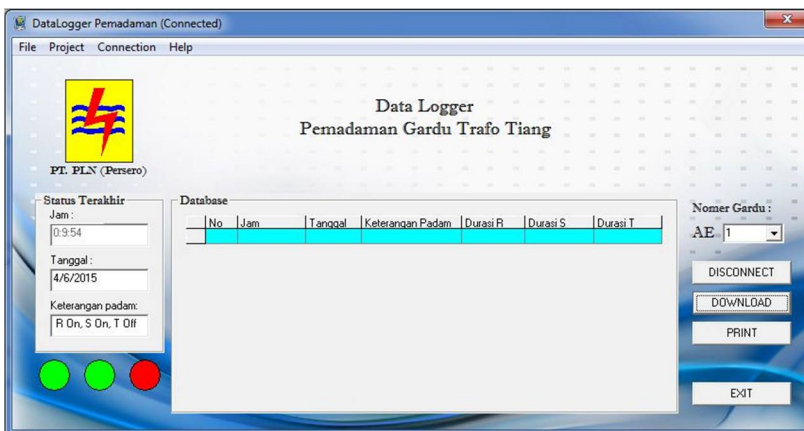


Gambar 4.14 Form Login pada HMI

4.6.2 Pengujian Tampilan Monitoring

Setelah berhasil melakukan proses *login*, maka proses selanjutnya ialah masuk ke tampilan utama data logger pemadaman gardu trafo tiang. Tampilan utama ini berisi tanggal, jam, keterangan padam serta durasi padam.

Saat masuk ke tampilan ini data-data tersebut belum terisi karena HMI belum terhubung dengan alat monitoring. Pada HMI ini terdapat 2 jenis pilihan untuk mendapatkan data logger pemadaman gardu. Yaitu dengan cara *online* dan dengan cara *offline*. Apabila secara *online*, petugas dapat meng klik menu *connect* dan data akan masuk secara otomatis ke HMI. Pada mode *online* dapat digunakan sebagai monitoring kondisi gardu trafo tiang tersebut dalam kondisi padam atau nyala. Sedangkan pada kondisi *offline*, meng klik tombol download untuk mendapatkan data frekuensi padam gardu trafo tiang dan durasi padam pada gardu trafo tiang yang dapat digunakan sebagai bahan perhitungan SAIFI (*System Average Interruption Frequency Index*) dan SAIDI (*System Average Interruption Duration Index*) yang berguna untuk menganalisa keandalan gardu trafo tiang. Gambar 4.15 menunjukkan tampilan HMI Data Logger Pemadaman GTT.

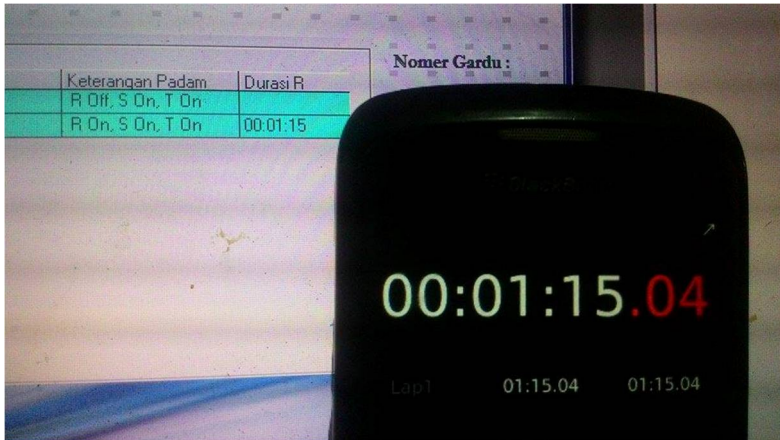


Gambar 4.15 Tampilan HMI Data Logger Pemadaman GTT

Seperti yang dapat dilihat pada gambar 4.12, pada HMI tertera waktu, tanggal, status padam, serta durasi padam yang terjadi. Selain dapat mengetahui durasi tiap kali fasa mati, pada HMI ini juga terdapat menu untuk mengetahui waktu serta frekuensi padam yang terjadi yang dapat diakses pada menu *project*.

4.7 Pengujian Penghitung Lama Waktu Padam

Karena pada tampilan *visual basic* di tampilkan durasi padam, maka dilakukan pengujian penghitung lama waktu padam menggunakan *stopwatch* untuk mengetahui ketepatan penghitung waktu pada program HMI. Gambar 4.16 Menunjukkan pengujian penghitung waktu



Gambar 4.16 Pengujian Penghitung Waktu

Pengujian dilakukan dengan cara mematikan salah satu fasa bersamaan dengan menjalankan *stopwatch*. Setelah beberapa menit, fasa yang mati tersebut dinyalakan kembali diikuti dengan menghentikan hitungan *stopwatch*. Kemudian hasil yang tertera di *stopwatch* dibandingkan dengan hasil yang tercatat pada HMI. Dari hasil pengujian didapatkan data seperti tabel 4.9 berikut :

Tabel 4.9 Pengujian Penghitung Waktu

Hasil <i>stopwatch</i>	Hasil Visual basic	Error (%)
00 : 00 : 14	00 : 00 : 14	0%
00 : 01 : 24	00 : 01 : 23	1,1%
00 : 02 : 27	00 : 02 : 27	0%
00 : 05 : 32	00 : 05 : 32	0%
00 : 12 : 53	00 : 12 : 52	1,1%

Perhitungan *error* pada tabel 4.9 di atas diperoleh dari persamaan berikut:

$$\% \text{ Error} = \frac{\text{Hasil perhitungan stopwatch} - \text{Hasil perhitungan HMI}}{\text{Hasil perhitungan stopwatch}} \times 100\%$$

Dari hasil perekaman waktu menggunakan visual basic maka dapat diketahui *error* penghitung lama waktu padam. Untuk *error* rata-rata pada data yang telah didapatkan maka dapat dihitung dengan rumus sebagai berikut :

$$e_r \text{ rata - rata} = \frac{\text{jumlah error}}{\text{banyaknya data yang diambil}}$$

Dari hasil pengujian penghitung lama waktu padam yang telah dilakukan, dapat disimpulkan pemakaian fitur ini memberikan *error* pengukuran sebesar 0.42%.

4.8 Pengujian Alat Keseluruhan

Keseluruhan kerja alat ini adalah ketika *power* dinyalakan, maka sumber menyuplai tegangan untuk rangkaian mikrokontroler, *Wiznet* dan *Wi-Fi* yang membutuhkan tegangan sebesar 5 Volt.

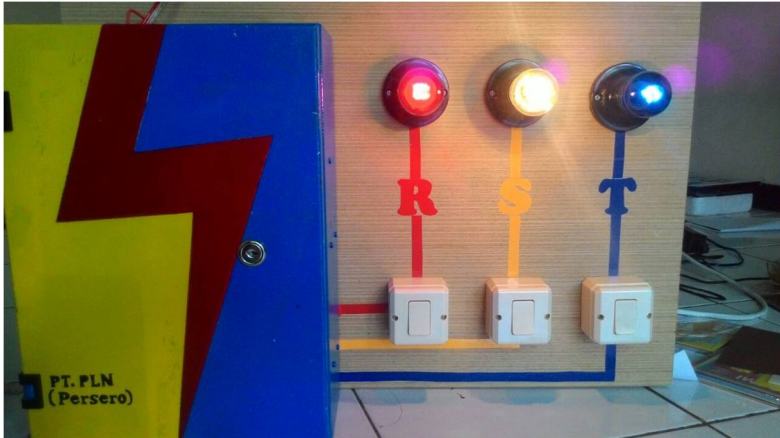
Mikrokontroler terhubung dengan rangkaian sensor tegangan dan komunikasi serial RS-232. Rangkaian sensor tegangan dihubungkan dengan *port* A.0 untuk fasa R, *port* A.1 untuk fasa S, *port* A.2 untuk fasa T pada mikrokontroler. Rangkaian RS-232 yang terdapat pada *port* C mikrokontroler dihubungkan dengan *Wiznet* yang berfungsi sebagai *Gateway* antara serial dengan LAN (*Local Area Network*).

Mikrokontroler diprogram untuk menerima data masukan dari sensor tegangan di *LV* panel. Kemudian mikrokontroler akan mengolah data dari sensor, sehingga dapat menentukan bahawa gardu dalam kondisi padam atau nyala dan mengirimkannya ke *PC server* dengan media *Wi-Fi* yang terdapat di *LV* panel ke jaringan *Wi-Fi* yang terdapat di *PC server*.

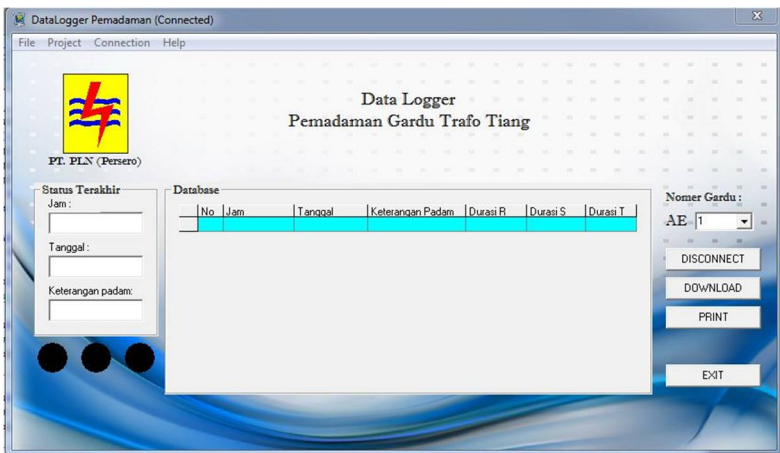
Pengujian dilakukan pada tujuh kondisi yaitu saat kondisi mula-mula normal, fasa R padam, S padam, T padam, R dan S padam, R dan T padam, S dan T padam, R S dan T padam, dan yang terakhir saat kembali normal.

4.8.1 Kondisi Normal Pertama Saat Alat Dinyalakan

Pengujian pertama dimulai pada kondisi saat semua fasa masih menyala. Gambar 4.17 menunjukkan lampu saat semua fasa menyala. Sedangkan gambar 4.18 menunjukkan tampilan HMI dimana saat pertama dinyalakan *data logger* masih kosong.



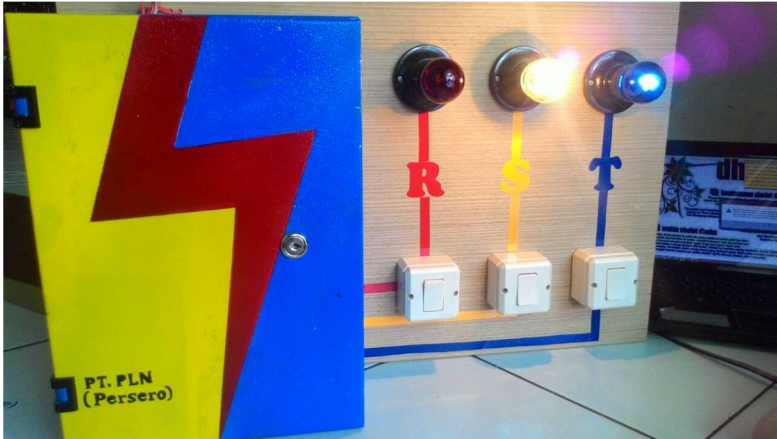
Gambar 4.17 Kondisi Awal saat Semua Fasa Normal



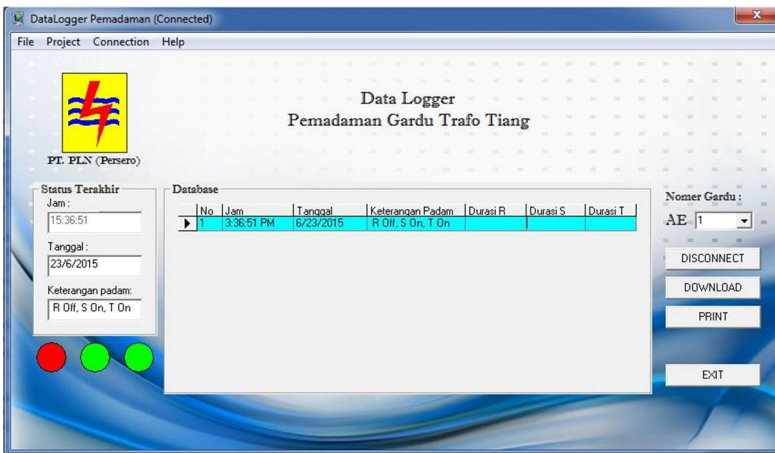
Gambar 4.18 Tampilan Awal HMI saat Semua Fasa Normal

4.8.2 Kondisi Fasa R Padam

Pengujian kedua dilakukan dengan kondisi fasa R padam. Gambar 4.19 menunjukkan indikator lampu fasa R mati sedangkan gambar 4.20 menunjukkan tampilan HMI saat fasa R padam.



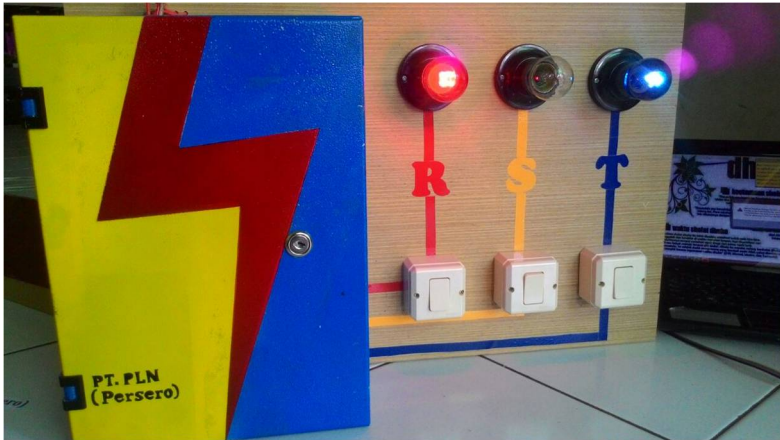
Gambar 4.19 Indikator Lampu Fasa R Mati



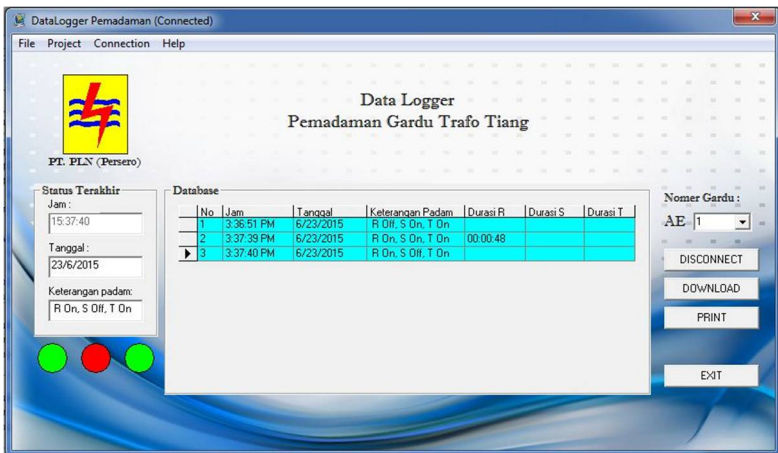
Gambar 4.20 Tampilan HMI Saat Fasa R Padam.

4.8.3 Kondisi Fasa S Padam

Pengujian ketiga dilakukan dengan kondisi fasa S padam. Gambar 4.21 menunjukkan indikator lampu fasa S mati sedangkan gambar 4.22 menunjukkan tampilan HMI saat fasa S padam.



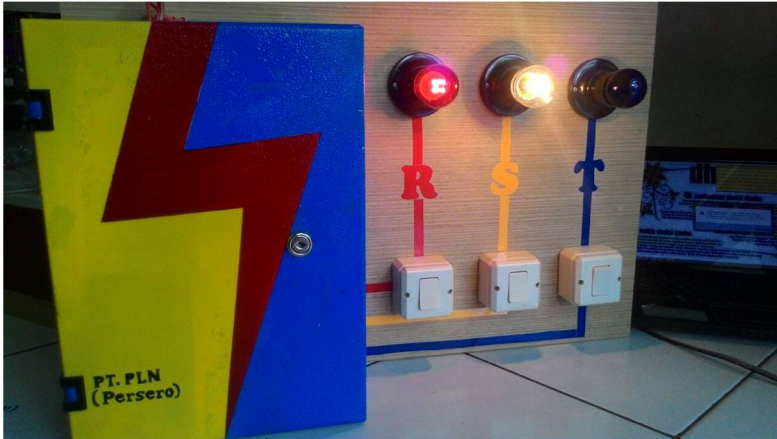
Gambar 4.21 Indikator Lampu Fasa S Mati



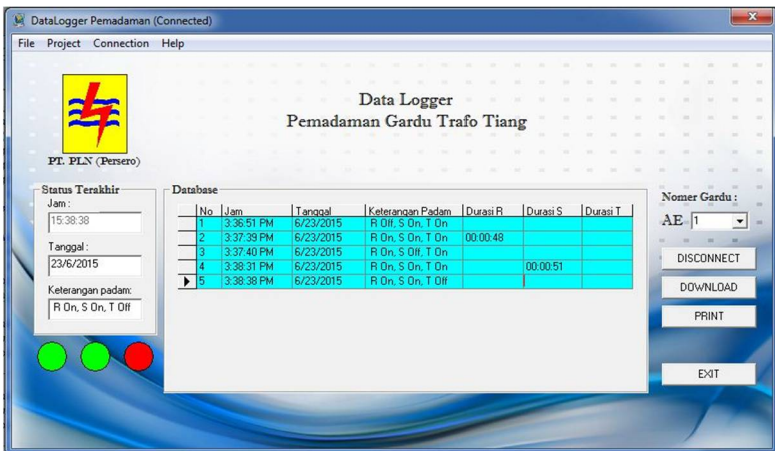
Gambar 4.22 Tampilan HMI saat Fasa S mati

4.8.4 Kondisi Fasa T Padam

Pengujian keempat dilakukan dengan kondisi fasa T padam. Gambar 4.23 menunjukkan indikator lampu fasa T mati sedangkan gambar 4.24 menunjukkan tampilan HMI saat fasa T padam.



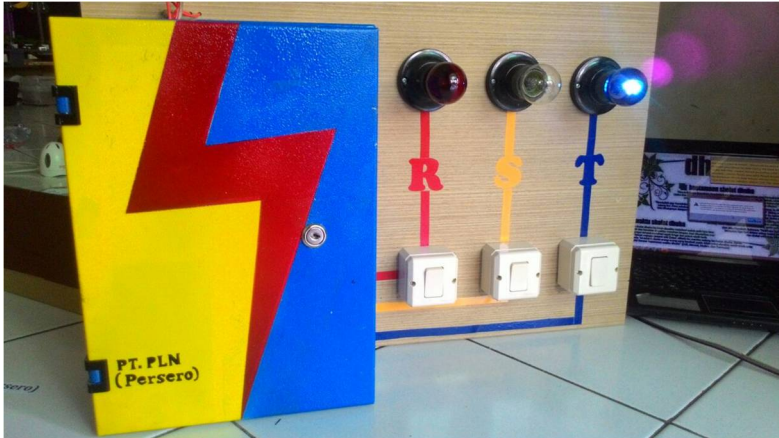
Gambar 4.23 Indikator Lampu Fasa T Mati



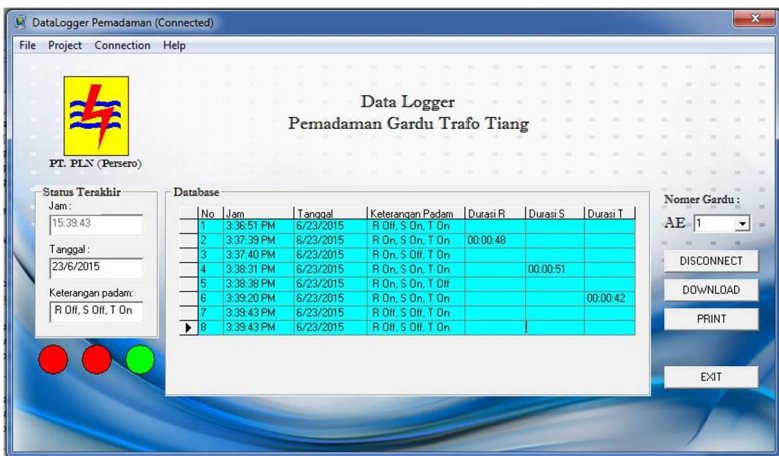
Gambar 4.24 Tampilan HMI Saat Fasa T Padam

4.8.5 Kondisi Fasa R dan S Padam

Pengujian kelima dilakukan dengan kondisi fasa R dan Spadam. Gambar 4.25 menunjukkan indikator lampu fasa R dan S mati sedangkan gambar 4.26 menunjukkan tampilan HMI saat fasa R dan S padam.



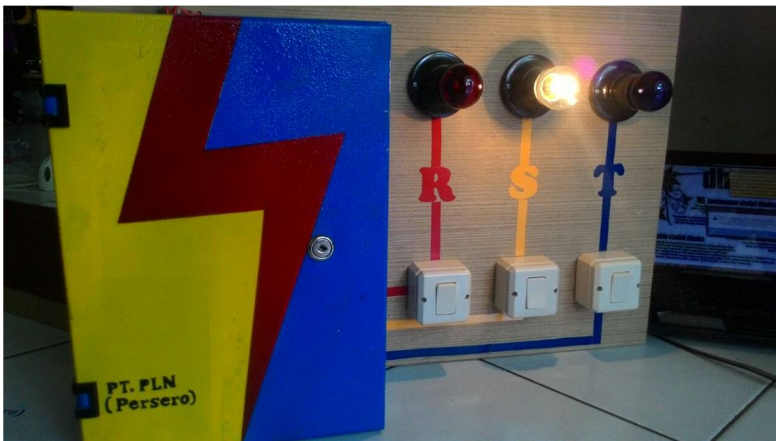
Gambar 4.25 Indikator Lampu Fasa R dan S Mati



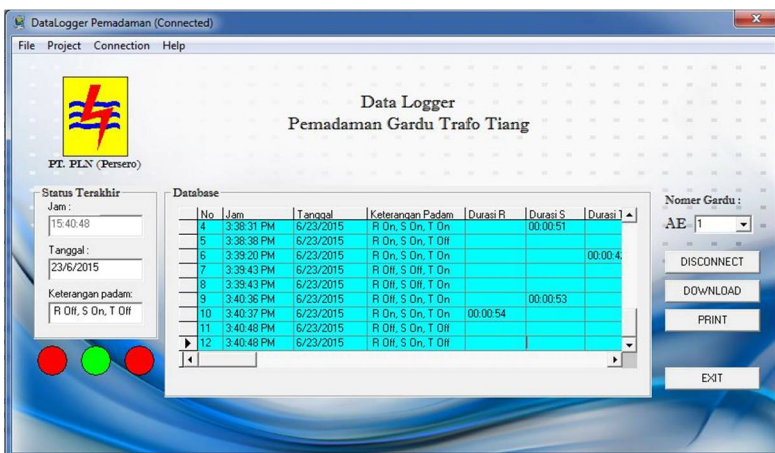
Gambar 4.26 Tampilan HMI Saat Fasa R dan S Padam

4.8.6 Kondisi Fasa R dan T Padam

Pengujian keenam dilakukan dengan kondisi fasa R dan T padam. Gambar 4.27 menunjukkan indikator lampu fasa R dan T mati sedangkan gambar 4.28 menunjukkan tampilan HMI saat fasa R dan T padam.



Gambar 4.27 Indikator Lampu Fasa R dan T Mati



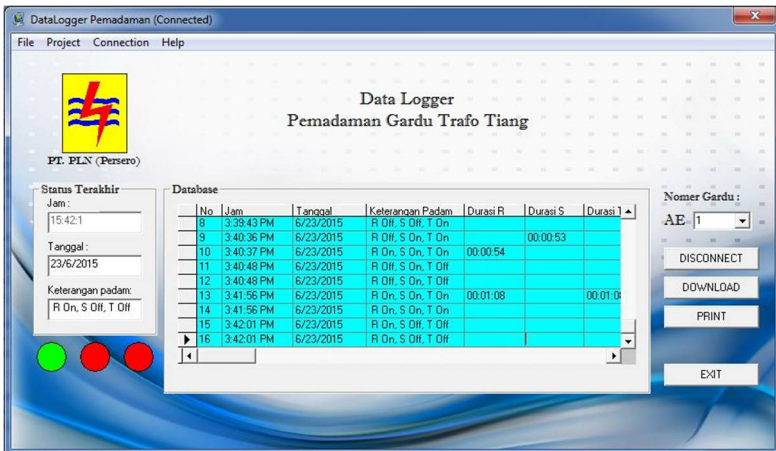
Gambar 4.28 Tampilan HMI Saat Fasa R dan T Padam

4.8.7 Kondisi Fasa S dan T Padam

Pengujian kedua dilakukan dengan kondisi fasa S dan T padam. Gambar 4.29 menunjukkan indikator lampu fasa S dan T mati sedangkan gambar 4.30 menunjukkan tampilan HMI saat fasa S dan T padam.



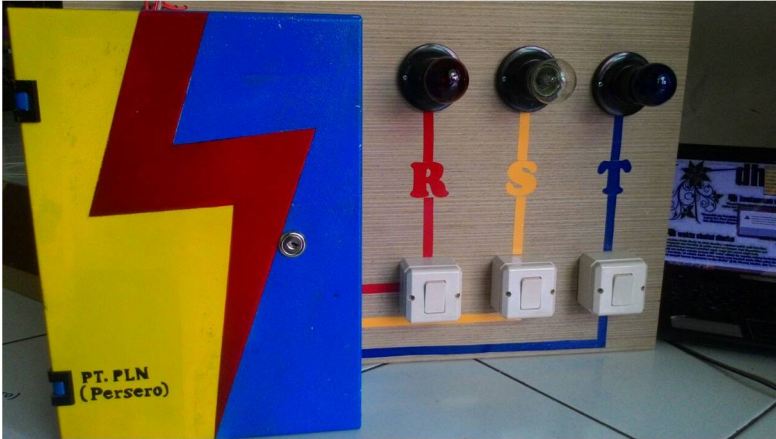
Gambar 4.29 Indikator Lampu Fasa S dan T Mati



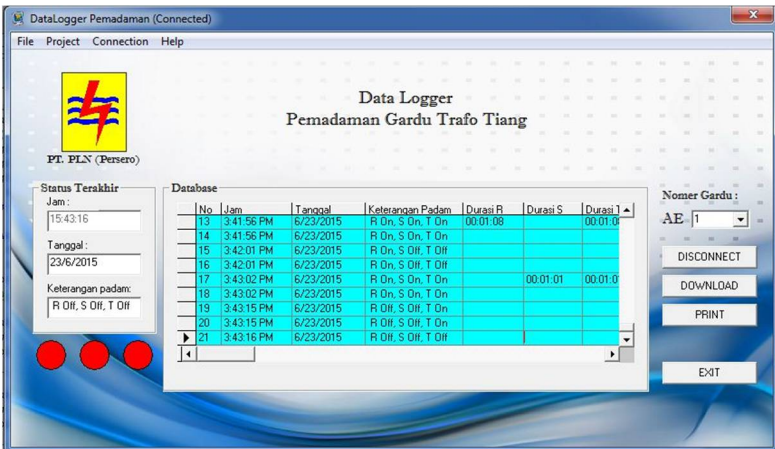
Gambar 4.30 Tampilan HMI Saat Fasa S dan T Padam

4.8.8 Kondisi Fasa R, S dan T Padam

Pengujian keenam dilakukan dengan kondisi fasa R, S, dan T padam. Gambar 4.31 menunjukkan indikator lampu fasa R, S dan T mati sedangkan gambar 4.32 menunjukkan tampilan HMI saat fasa R,S dan T padam.



Gambar 4.31 Indikator Lampu Fasa R, S dan T Mati



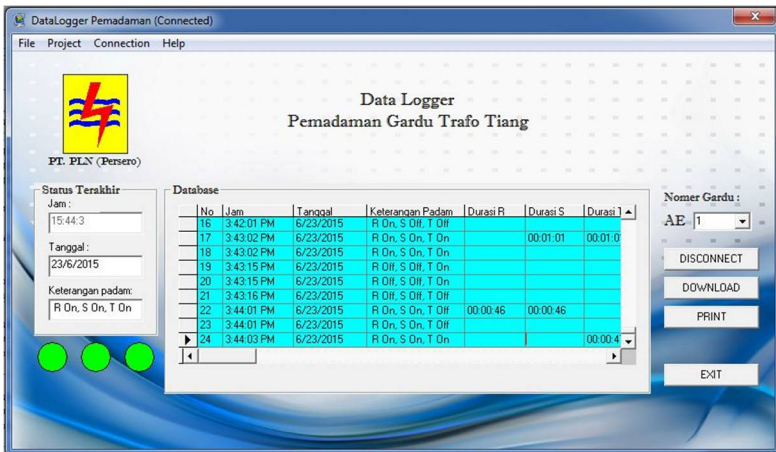
Gambar 4.32 Tampilan HMI Saat Fasa R, S dan T Padam

4.8.9 Kondisi Semua Fasa Kembali Normal

Pengujian terakhir dilakukan dengan kondisi semua fasa kembali menyala. Gambar 4.33 menunjukkan semua indikator lampu fasa menyala sedangkan gambar 4.34 menunjukkan tampilan HMI saat semua fasa kembali normal.

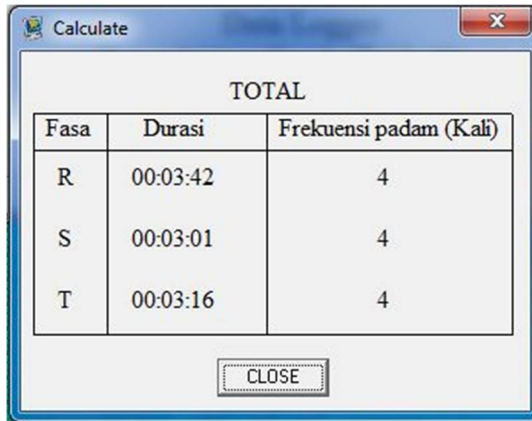


Gambar 4.33 Indikator Lampu Fasa R, S dan T Kembali Menyala



Gambar 4.34 Tampilan HMI Saat Fasa R, S dan T Kembali Menyala

Selain dapat mengetahui waktu, tanggal, serta durasi setiap kali padam, pada HMI juga disediakan menu untuk mengetahui total waktu serta frekuensi padam tiap fasa. Untuk mengetahui waktu dan frekuensi padam total tiap fasa, dapat dipilih pada menu *project* lalu klik *calculate*. Gambar 4.35 menunjukkan total waktu dan fekuensi padam tiap fasa pada pengujian alat keseluruhan di atas.



TOTAL		
Fasa	Durasi	Frekuensi padam (Kali)
R	00:03:42	4
S	00:03:01	4
T	00:03:16	4

CLOSE

Gambar 4.35 Total Waktu dan Frekuensi Padam

Data keterangan padam yang tersimpan pada *microsoft access* dapat ditampilkan melalui aplikasi *crystal report* dengan meng-klik tombol '*print*' pada HMI. Gambar 4.36 menunjukkan hasil rekap data keterangan padam.

Setelah pengujian alat keseluruhan didapatkan hasil bahwa data yang dikirim mikrokontroler melalui *Wi-Fi* ke komputer, terkadang tidak semuanya dapat diterima dan sebagian ada data yang hilang. Hal ini disebabkan karena antar mikrokonter dan komputer tidak selaras dan juga kendala jarak antara *Wi-Fi* (*Wireless Fidelity*) dengan komputeryang terhalang oleh banyak penghalang. Namun kebanyakan percobaan yang kami lakukan sedikit mengalami kendala tersebut. Kendala lain yang kami temui adalah bila ada data yang hilang maka penyimpanan ke database komputer akan mengalami *error* sehingga aplikasi HMI akan terhenti dan terpaksa kita harus mulai ulang aplikasi HMI dan mereset mikrokontrolernya. Agar data tetap tersimpan, maka *data logger* dilengkapi *baterry backup* yang berguna untuk *supply* cadangan bila terjadi padam. Sehingga data yang tersimpan tidak hilang.



LAPORAN PEMADAMAN
GARDU TRAF0 TIANG

Jl. Wiratno No.53 . Surabaya



No. Gardu AE 001

No	Jam	Tanggal	Keterangan Padam	Durasi R	Durasi S	Durasi T
1	3:36:51 pm	06/23/2015	R Off, S On, T On			
2	3:37:39 pm	06/23/2015	R On, S On, T On	00:00:48		
3	3:37:40 pm	06/23/2015	R On, S Off, T On			
4	3:38:31 pm	06/23/2015	R On, S On, T On		00:00:51	
5	3:38:38 pm	06/23/2015	R On, S On, T Off			
6	3:39:20 pm	06/23/2015	R On, S On, T On			00:00:42
7	3:39:43 pm	06/23/2015	R Off, S Off, T On			
8	3:39:43 pm	06/23/2015	R Off, S Off, T On			
9	3:40:36 pm	06/23/2015	R Off, S On, T On		00:00:53	
10	3:40:37 pm	06/23/2015	R On, S On, T On	00:00:54		
11	3:40:48 pm	06/23/2015	R Off, S On, T Off			
12	3:40:48 pm	06/23/2015	R Off, S On, T Off			
13	3:41:56 pm	06/23/2015	R On, S On, T On	00:01:08		00:01:08
14	3:41:56 pm	06/23/2015	R On, S On, T On			
15	3:42:01 pm	06/23/2015	R On, S Off, T Off			
16	3:42:01 pm	06/23/2015	R On, S Off, T Off			
17	3:43:02 pm	06/23/2015	R On, S On, T On		00:01:01	00:01:01
18	3:43:02 pm	06/23/2015	R On, S On, T On			
19	3:43:15 pm	06/23/2015	R Off, S Off, T On			
20	3:43:15 pm	06/23/2015	R Off, S Off, T On			
21	3:43:16 pm	06/23/2015	R Off, S Off, T Off			
22	3:44:01 pm	06/23/2015	R On, S On, T Off	00:00:46	00:00:46	
23	3:44:01 pm	06/23/2015	R On, S On, T Off			
24	3:44:03 pm	06/23/2015	R On, S On, T On			00:00:47

Gambar 4.36 Hasil Rekap Data Keterangan Padam

4.9 Analisa Relevansi

Data logger pemadaman gardu trafo tiang ini dapat diterapkan dalam kehidupan sehari-hari dengan catatan hal-hal yang belum sempurna dalam *prototype* telah disesuaikan dengan *LV panel* yang sebenarnya. Apabila *data logger* ini diterapkan, maka akan meningkatkan kualitas pelayanan dan meningkatkan kehandalan sistem penyaluran energi listrik PLN karena petugas pelayanan teknik dapat mengetahui kondisi padam gardu trafo tiang tanpa harus menunggu mendapat laporan padam dari konsumen.

Selain itu penerapan alat ini dapat mengurangi waktu padam listrik, sehingga masyarakat dapat memanfaatkan energi listrik dengan

semestinya. Karena dengan adanya alat ini, PLN dapat mengurangi laporan keluhan pelanggan, serta memangkas waktu petugas terjun ke lapangan dan menyusuri lokasi gangguan dan mengidentifikasi letak penyebab gangguan tersebut.

BAB V

PENUTUP

5.1 Kesimpulan

Dari Tugas Akhir yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Data logger pemadaman gardu trafo tiang (GTT) ini dapat mencatat waktu padam, durasi padam, dan frekuensi padam serta menjadi alat monitoring untuk mengetahui kondisi gardu apakah padam atau tidak.
2. Nilai pengujian sensor tegangan mempunyai presentase *error* masing-masing fasa kurang dari 2%.
3. Nilai pengujian *EEPROM* mempunyai presentase *error* masing-masing fasa sebesar 0.42%.
4. Mikrokontroler dapat mengirimkan data pada *EEPROM* berupa waktu, tanggal, keterangan padam dengan selang 1 detik terhadap data yang akan dikirim selanjutnya.
5. Dari pengujian padam sebanyak 7 kali, data dapat terekam semuanya selama alat masih dalam jangkauan *wi-fi*.
6. Alat data logger pemadaman gardu trafo tiang bekerja pada *range* tegangan 170-240 VAC, membutuhkan arus sebesar 20 mA. Sensor tegangan dapat mengukur tegangan dari *range* 0 sampai dengan 230 VAC.

5.2 Saran

Dengan memperhatikan beberapa kelemahan dan kekurangan dari proyek Tugas Akhir ini, maka diberikan beberapa saran yang sekiranya dapat dikembangkan pada masa yang akan datang demi kesempurnaan dari proyek Tugas Akhir ini. Adapun beberapa saran tersebut yaitu:

1. Untuk mengembangkan kapasitas penyimpanan pada *data logger* maka *EEPROM* yang berfungsi sebagai memori dapat diganti dengan menggunakan *SD card* yang berkapasitas lebih besar daripada *EEPROM*.
2. Bila datalogger digunakan sebagai alat monitoring gardu, maka *Wi-Fi (Wireless Fidelity)* yang terdapat pada alat tersebut dapat diganti dengan *Wi-Fi* yang mempunyai jangkauan lebih jauh.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

1., Data Logger, <<http://sonoku.com/data-logger-bagian-1/>>, 25 Mei 2015
2., *SPLN: Buku 4: Standar Konstruksi Gardu Distribusi & Gardu Hubung Tenaga*, PT.PLN (Persero), Jakarta. 2010.
3. Nurcahyo, Sidik.. *Aplikasi dan Teknik Pemrograman Mikrokontroler AVR ATMEL*. Andi : Yogyakarta. 2012.
4., Real Time Clock, < <https://learn.adafruit.com/ds1307-real-time-clock-breakout-board-kit/understanding-the-code>>, 25 Mei 2015.
5., RS 232, < <http://www.arcelect.com/rs232.htm>>, 25 Mei 2015.
6., WIZ110SR User Manual, WIZnet, Korean, 1998.
7., Wi-fi, <<https://id.wikipedia.org/wiki/Wi-Fi>>, 25 Mei 2015.
8. Dewobrata, Wiryanto. *Aplikasi Saindan Teknik dengan Visual Basic 6.0*. PT. elex Media Komputindo : Yogyakarta. 2002.

Halaman ini sengaja dikosongkan

LAMPIRAN 1

1. Listing Program Mikrokontroler

```
#include <mega16.h>

#include <delay.h>
#include <stdlib.h>
#include <stdio.h>
#define EEPROM_BUS_ADDRESS 0x50

int terima=0;

// I2C Bus functions
#asm
    .equ __i2c_port=0x15 ;PORTC
    .equ __sda_bit=1
    .equ __scl_bit=0
#endasm
#include <i2c.h>

// DS1307 Real Time Clock functions
#include <ds1307.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
```

```

#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE<256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine

```

```

interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index]=data;
if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;
rx_buffer_overflow=1;
};
};
lcd_clear();
if(data=='A')
{
terima = 1;
}
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#asm("cli")
--rx_counter;
#asm("sei")
return data;
}
#pragma used-

```

```

#endif

// Standard Input/Output functions
#include <stdio.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

// Declare your global variables here
char temp[16];
int r,s,t;
int addrsl=1;
int looping=0;
int addr=1;
int cek=1;
char temp[16],buff[16],buffer[16];
unsigned char h, m, sc, d, mo, y;

void tampilan_sensor()
{
    sprintf(temp,"R:%i",r);
    lcd_gotoxy(0,0);
    lcd_puts(temp);

    sprintf(buffer,"S:%i",s);

```

```

    lcd_gotoxy(0,1);
    lcd_puts(buffer);

    sprintf(buff, "T:%i",t);
    lcd_gotoxy(9,0);
    lcd_puts(buff);
}

void baca_sensor()
{
    r=((0.2846* read_adc(0)) + 3.9418);
    delay_ms(5);
    s=((0.2948 * read_adc(1))-1.9306);
    delay_ms(5);
    t=((0.3037 * read_adc(2))-9.9096);
    delay_ms(5);
}

void akses_rtc()
{
    rtc_get_time(&h,&m,&sc); // mengakses jam
    rtc_get_date(&d,&mo,&y); //mengakses tanggal
}

void tampilan_rtc()
{
    lcd_gotoxy(0,0);
    sprintf(temp, "Time %d:%d:%d",h,m,sc);
    lcd_puts(temp);
    lcd_gotoxy(0,1);
    sprintf(temp, "Date %d-%d-%d",d,mo,y);
    lcd_puts(temp);
    delay_ms(200);
    lcd_clear();
}

unsigned char eeprom_read(unsigned char address)
{
    unsigned char data;

```

```

i2c_start();
i2c_write(EEPROM_BUS_ADDRESS & 0xFE);
i2c_write(address);
i2c_start();
i2c_write(EEPROM_BUS_ADDRESS | 1);
data=i2c_read(0);
i2c_stop();
return data;
}

```

```

void eeprom_write(unsigned char address, unsigned char data)
{
    i2c_start();
    i2c_write(EEPROM_BUS_ADDRESS & 0xFE);
    i2c_write(address);
    i2c_write(data);
    i2c_stop();
    delay_ms(10);
}

```

```

void write_to_eeprom()
{
    if(addr < 2701)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts("Write to EEPROM");
        eeprom_write(addr,h);
        addr++;
        delay_ms(25);
        eeprom_write(addr,m);
        addr++;
        delay_ms(25);
        eeprom_write(addr,sc);
        addr++;
        delay_ms(25);
        eeprom_write(addr,d);
        addr++;
        delay_ms(25);
    }
}

```

```

    eeprom_write(addr,mo);
    addr++;
    delay_ms(25);
    eeprom_write(addr,y);
    addr++;
    delay_ms(25);
    eeprom_write(addr,r);
    addr++;
    delay_ms(25);
    eeprom_write(addr,s);
    addr++;
    delay_ms(25);
    eeprom_write(addr,t);
    addr++;
    delay_ms(25);
}else
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_puts(" EEPROM Penuh ");
    delay_ms(500);
}
}

```

```

void read_from_eeprom()
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_puts(" Sending... ");
    printf("#%i",eeprom_read(addr1));
    addr1++;
    delay_ms(25);
    printf(":%i",eeprom_read(addr1));
    addr1++;
    delay_ms(25);
    printf(":%i",eeprom_read(addr1));
    addr1++;
    delay_ms(25);
    printf("#%i",eeprom_read(addr1));
}

```



```

    addrs1++;
    delay_ms(25);
    printf("/%i",eeprom_read(addrs1));
    addrs1++;
    delay_ms(25);
    printf("/20%i",eeprom_read(addrs1));
    addrs1++;
    delay_ms(25);
    printf("#%i",eeprom_read(addrs1));
    addrs1++;
    delay_ms(25);
    printf("#%i",eeprom_read(addrs1));
    addrs1++;
    delay_ms(25);
    printf("#%i",eeprom_read(addrs1));
    addrs1++;
    delay_ms(25);
}

void main(void)
{
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
    Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTA=0x00;
    DDRA=0x00;

    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
    Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTB=0x00;
    DDRB=0x00;
}

```

```
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;
```

```
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
```

```

TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;

```

```

UBRR=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIO=0x00;

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: Free Running
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0xA4;
SFIO&=0x1F;

// I2C Bus initialization
i2c_init();

// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: Off
// SQW/OUT pin state: 0
rtc_init(0,0,0);

// LCD module initialization
lcd_init(16);

// Global enable interrupts
asm("sei")

lcd_gotoxy(0,0);
lcd_puts(" Data Logger ");
lcd_gotoxy(0,1);
lcd_puts(" Pemadaman GTT ");
delay_ms(1000);
lcd_clear();

while (1)
{

```

```

    baca_sensor();
    akses_rtc();
    tampilan_rtc();
    if (terima == 1)
    {
        for (looping = 0; looping < ((addrs-1)/9); looping++)
        {
            read_from_eeprom();
            delay_ms(750);
        }
        terima=0;
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts(" Succesfully ");
        delay_ms(250);
    };

        if (r<150)
    {
        if(cek==1)
        {
            write_to_eeprom();
            addrs1 = addrs - 9;
            read_from_eeprom();
            cek=0;
        }
    }else
    {
        if (cek == 0)
        {
            write_to_eeprom();
            addrs1 = addrs - 9;
            read_from_eeprom();
            cek=1;
        }
    }

    if (s < 150)
    {

```

```

if(cek1==1)
{
    write_to_eeprom();
    addrsl = addrsl - 9;
    read_from_eeprom();
    cek=0;
}
}else
{
    if (cek1 == 0)
    {
        write_to_eeprom();
        addrsl = addrsl - 9;
        read_from_eeprom();
        cek=1;
    }
}

if (t<150)
{
    if(cek2==1)
    {
        write_to_eeprom();
        addrsl = addrsl - 9;
        read_from_eeprom();
        cek=0;
    }
}else
{
    if (cek2 == 0)
    {
        write_to_eeprom();
        addrsl = addrsl - 9;
        read_from_eeprom();
        cek=1;
    }
}
};
}

```

2. Listing Program Visual Basic 6.0

```
Dim strArus As String
Dim pesan() As String
Dim valteg1 As Double
Dim valteg2 As Double
Dim valteg3 As Double
Dim detik As Integer
Dim dtk As Integer
Dim a
Dim frek As Integer
Dim frek1 As Integer
Dim frek2 As Integer
Option Explicit
'Bismillah, semoga lancar amiin!!
```

```
Private Sub About_Click()
Form3.Show
Form1.Enabled = False
End Sub
```

```
Private Sub Calculate_Click()
Form4.Show
Form1.Enabled = False
End Sub
```

```
Private Sub cmdConnect_Click()
If cmdConnect.Caption = "CONNECT" Then
    If Val(Combo1) = 1 Then
        Winsock1.Close
        Winsock1.RemoteHost = "192.168.11.2"
        Winsock1.RemotePort = "5000"
        Winsock1.Connect
    Else
        MsgBox ("Nomer gardu salah")
    End If
Else
    If MsgBox("Do you want to disconnect from server. . .?",
vbQuestion + vbOKCancel, "Confirm") = vbOK Then
```

```

Winsock1.Close
Form1.Caption = "DataLogger Pemadaman (Disconnected)"
cmdConnect.Caption = "CONNECT"
Connect.Caption = "Connect"
CmdDownload.Enabled = False
Download.Enabled = False
Connect.Enabled = True
End If
End If
End Sub

```

```

Private Sub CmdDownload_Click()
Text17.Text = ""
Text18.Text = ""
Text19.Text = ""
Text20.Text = "00:00:00"
Text21.Text = "00:00:00"
Text22.Text = "00:00:00"
frek = 0
frek1 = 0
frek2 = 0
If Adodc1.Recordset.RecordCount > 0 Then
    *** hapus record di db
    With Adodc1.Recordset
        .MoveFirst
        Do
            .Delete
            .MoveNext
        Loop Until .EOF

    End With
End If
Winsock1.SendData "A"
End Sub

```

```

Private Sub cmdPrint_Click()
With CReport1
    .RetrieveDataFiles
    .PrintReport

```



```
End With  
End Sub
```

```
Private Sub Command1_Click()  
If (MsgBox("Are you sure want to exit?", vbYesNo, "Exit") = vbYes)  
Then  
    Unload Me  
End If  
End Sub
```

```
Private Sub Connect_Click()  
If cmdConnect.Caption = "CONNECT" Then  
    If Val(Combo1) = 1 Then  
        Winsock1.Close  
        Winsock1.RemoteHost = "192.168.11.2"  
        Winsock1.RemotePort = "5000"  
        Winsock1.Connect  
    Else  
        MsgBox ("Nomer gardu salah")  
    End If  
    Else  
        If MsgBox("Do you want to disconnect from server. . .?",  
vbQuestion + vbOKCancel, "Confirm") = vbOK Then  
            Winsock1.Close  
            Form1.Caption = "DataLogger Pemadaman (Disconnected)"  
            cmdConnect.Caption = "CONNECT"  
            Connect.Caption = "Connect"  
            CmdDownload.Enabled = False  
            Download.Enabled = False  
            Connect.Enabled = True  
        End If  
    End If  
End Sub
```

```
Private Sub Download_Click()  
Text17.Text = ""  
Text18.Text = ""
```

```

Text19.Text = ""
Text20.Text = "00:00:00"
Text21.Text = "00:00:00"
Text22.Text = "00:00:00"
frek = 0
frek1 = 0
frek2 = 0
If Adodc1.Recordset.RecordCount > 0 Then
    *** hapus record di db
    With Adodc1.Recordset
        .MoveFirst
        Do
            .Delete
            .MoveNext
        Loop Until .EOF

    End With
End If
Winsock1.SendData "A"
End Sub

Private Sub Exit_Click()
If (MsgBox("Are you sure want to exit?", vbYesNo, "Exit") = vbYes)
Then
    Unload Me
End If
End Sub

Private Sub Form_Load()
Form2.Show
Form1.Hide
CmdDownload.Enabled = False
Download.Enabled = False
Timer2.Enabled = False
CReport1.ReportFileName = App.Path & "\Database.rpt"
Adodc1.RecordSource = App.Path & "\Database.mdb"
'Skin1.LoadSkin App.Path + "\Le-Black.skn"
'Skin1.ApplySkin Me.hWnd
For a = 1 To 40

```

```
    Combo1.AddItem (a)
Next a
```

```
End Sub
```

```
Private Sub Form_Resize()
Image2.Width = Width
Image2.Height = Height
End Sub
```

```
Private Sub Print_Click()
With CReport1
    .RetrieveDataFiles
    .PrintReport
End With
End Sub
```

```
Private Sub Text1_Change()
On Error Resume Next
Dim tot As Integer
Dim Tanggal As String
Dim pager() As String
If Text1.Text <> "" Then
'temp = Text1.Text

End If
```

```
If Trim(Text1.Text) <> "" Then
pesan = Split(Text1.Text, "#")
tot = UBound(pesan)
```

```
If tot > 0 Then
'Tanggal = pesan(1)
```

```
Text2.Text = pesan(1)
Text3.Text = pesan(2)
Text4.Text = pesan(3)
Text5.Text = pesan(4)
```

```
Text6.Text = pesan(5)
'strTegangan = pesan(3)
'strArus = pesan(2)
'Text4.Text = strArus
'Tegangan.Text = strTegangan
```

```
Timer2.Enabled = True
```

```
End If
End If
End Sub
```

```
Private Sub Text12_Change()
If Text12.Text <> "00:00:00" Then
Text13.Text = totalWaktu(Text11, Text12)
End If
End Sub
```

```
Private Sub Text15_Change()
If Text15.Text <> "00:00:00" Then
Text16.Text = totalWaktu(Text14, Text15)
End If
End Sub
```

```
Private Sub Text9_Change()
If Text9.Text <> "00:00:00" Then
Text10.Text = totalWaktu(Text8, Text9)
End If
End Sub
```

```
Private Sub Timer2_Timer()
dtk = dtk + 1
If dtk = 2 Then
Text7.Text = ""
valteg1 = Val(Text4.Text)
valteg2 = Val(Text5.Text)
valteg3 = Val(Text6.Text)
```

```
If valteg1 < 190 Then
```

```

Text7.Text = Text7.Text & " R Off"
Shape1.FillColor = &HFF
If Text8.Text = "00:00:00" Then
Text8.Text = Text2.Text
End If
'Text13.Text = ""
'Text10.Text = ""
Else
Shape1.FillColor = vbGreen
Text7.Text = Text7.Text & " R On"
If Text8.Text <> "00:00:00" And Text9.Text = "00:00:00" Then
Text9.Text = Text2.Text
End If
End If

```

```

If valteg2 < 190 Then
Text7.Text = Text7.Text & ", S Off"
Shape2.FillColor = &HFF
If Text11.Text = "00:00:00" Then
Text11.Text = Text2.Text
End If
'Text10.Text = ""
'Text13.Text = ""
Else
Shape2.FillColor = vbGreen
Text7.Text = Text7.Text & ", S On"
If Text11.Text <> "00:00:00" And Text12.Text = "00:00:00" Then
Text12.Text = Text2.Text
End If
End If

```

```

If valteg3 < 190 Then
Text7.Text = Text7.Text & ", T Off"
Shape3.FillColor = &HFF
If Text14.Text = "00:00:00" Then
Text14.Text = Text2.Text
End If
'Text8.Text = Text2.Text

```

```

Text10.Text = ""
Else
Shape3.FillColor = vbGreen
Text7.Text = Text7.Text & ", T On"
If Text14.Text <> "00:00:00" And Text15.Text = "00:00:00" Then
Text15.Text = Text2.Text
End If
End If

```

```

'If valteg1 > 190 And valteg2 > 190 And valteg3 > 190 Then
'Text7.Text = "Normal"
'Shape1.FillColor = vbGreen
'Text9.Text = Text2.Text
'End If

```

```

'If Text10.Text = "" Then
'Text12.Text = "00:00:00"
'Else
'If Text10.Text <> "" Then
'Text13.Text = jumlahWaktu(Text13, Text10)
'End If
'End If
'If Text12.Text <> "" And Text13.Text <> "" Then
'Text14.Text = jumlahWaktu(Text12, Text13)
'Text12.Text = ""
'Text13.Text = ""
'End If

```

```

'If Shape1.FillColor = &HFF Then 'menghitung frekuensi padam
'frek = frek + 1
'Text11.Text = frek
'End If

```

```

Adodc1.Recordset.AddNew 'menulis ke database
Me.Adodc1.Recordset.Fields(0) = Adodc1.Recordset.RecordCount
Me.Adodc1.Recordset.Fields(1) = Text2.Text 'kolom 1 berisi text 2
yaitu jam

```

```

Me.Adodc1.Recordset.Fields(2) = Text3.Text    'kolom 2 berisi text 3
yaitu tanggal
Me.Adodc1.Recordset.Fields(3) = Text7.Text    'kolom 3 berisi text7
yaitu keterangan padam
Me.Adodc1.Recordset.Fields(4) = Text10.Text    'kolom 4 berisi text 10
yaitu durasi
Me.Adodc1.Recordset.Fields(5) = Text13.Text
Me.Adodc1.Recordset.Fields(6) = Text16.Text
'Me.Adodc1.Recordset.Fields(6) = Text10.Text
Adodc1.Recordset.Update

```

```

If Text10.Text <> "" Then
Text8.Text = "00:00:00"
Text9.Text = "00:00:00"
Text20.Text = jumlahWaktu(Text20, Text10)
Text10.Text = ""
frek = frek + 1
Text17.Text = frek
End If

```

```

If Text13.Text <> "" Then
Text11.Text = "00:00:00"
Text12.Text = "00:00:00"
Text21.Text = jumlahWaktu(Text21, Text13)
Text13.Text = ""
frek1 = frek1 + 1
Text18.Text = frek1
End If

```

```

If Text16.Text <> "" Then
Text14.Text = "00:00:00"
Text15.Text = "00:00:00"
Text22.Text = jumlahWaktu(Text22, Text16)
Text16.Text = ""
frek2 = frek2 + 1
Text19.Text = frek2
End If

```

```

Text1.Text = ""

```

```
dtk = 0
Timer2.Enabled = False
End If
```

```
If dtk = 3 Then
```

```
End If
End Sub
```

```
Private Sub Winsock1_Close()
Form1.Caption = "DataLogger Pemadaman (Disconnected)"
cmdConnect.Caption = "CONNECT"
Connect.Caption = "Connect"
CmdDownload.Enabled = False
Download.Enabled = False
Connect.Enabled = True
End Sub
```

```
Private Sub Winsock1_Connect()
If Winsock1.State = sckConnected Then
    Form1.Caption = "DataLogger Pemadaman (Connected)"
    MsgBox ("connected")
    cmdConnect.Caption = "DISCONNECT"
    Connect.Caption = "Disconnect"
    CmdDownload.Enabled = True
    Download.Enabled = True
    Connect.Enabled = True
End If
End Sub
```

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim strData As String
'get the data from the socket
Winsock1.GetData strData
'display it in the textbox
Text1.Text = Text1.Text & strData
'scroll the box down
Text1.SelStart = Len(Text1.Text)
End Sub
```



```

Private Sub Winsock1_Error(ByVal Number As Integer, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile
As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
'an error has occurred somewhere,so let the user know
MsgBox "Error:" & Description
'close the socket, ready to go again
Winsock1.Close
End Sub
Private Function totalWaktu(jamAwal As Variant, jamAkhir As
Variant) As String

```

```

Dim detikAkhir, detikAwal, jumlahDetik As Long

```

```

detikAwal = (Hour(jamAwal) * 3600) + (Minute(jamAwal) * 60) +
(Second(jamAwal))

```

```

detikAkhir = (Hour(jamAkhir) * 3600) + (Minute(jamAkhir) * 60) +
(Second(jamAkhir))

```

```

If jamAkhir < jamAwal Then
'---di hitung satu hari
jumlahDetik = 86400
Else
jumlahDetik = 0
End If

```

```

jumlahDetik = jumlahDetik + (detikAkhir - detikAwal)

```

```

totalWaktu = Format(Str(Int((Int((jumlahDetik / 3600)) Mod 24))),
"00") + _
":" + Format(Str(Int((Int((jumlahDetik / 60)) Mod 60))), "00") + _
":" + Format(Str(Int((jumlahDetik Mod 60))), "00")

```

```

End Function

```

```

Private Function jumlahWaktu(jamAwal As Variant, jamAkhir As
Variant) As String

```

Dim detikAkhir, detikAwal, jumlahDetik As Long

detikAwal = (Hour(jamAwal) * 3600) + (Minute(jamAwal) * 60) +
(Second(jamAwal))

detikAkhir = (Hour(jamAkhir) * 3600) + (Minute(jamAkhir) * 60) +
(Second(jamAkhir))

'If jamAkhir < jamAwal Then

'----di hitung satu hari

'jumlahDetik = 86400

'Else

'jumlahDetik = 0

'End If

jumlahDetik = (detikAkhir + detikAwal)

jumlahWaktu = Format(Str(Int((Int((jumlahDetik / 3600))))), "00") + _
":" + Format(Str(Int((Int((jumlahDetik / 60)) Mod 60))), "00") + _
":" + Format(Str(Int((jumlahDetik Mod 60))), "00")

End Function

Halaman ini sengaja dikosongkan

LAMPIRAN 2

LISTING DATASHEET

1. Datasheet ATmega16

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 16K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 512 Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 1K Byte Internal SRAM
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega16L
 - 4.5 - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16



8-bit AVR[®]
Microcontroller
with 16K Bytes
In-System
Programmable
Flash

ATmega16
ATmega16L

Preliminary

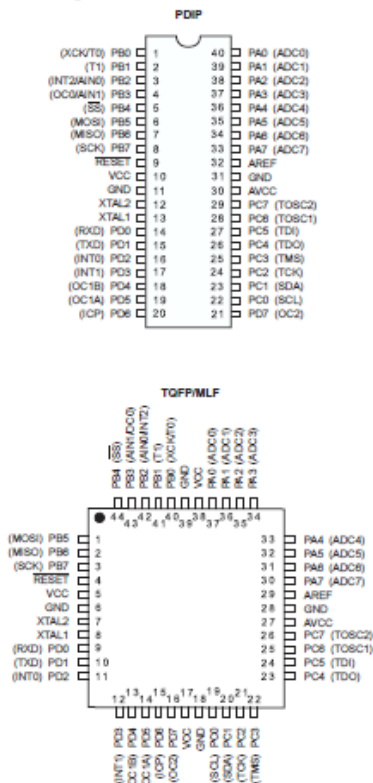
Rev. 2480E-AVR-10/02





Pin Configurations

Figure 1. Pinouts ATmega16



Disclaimer

Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

2 ATmega16(L)

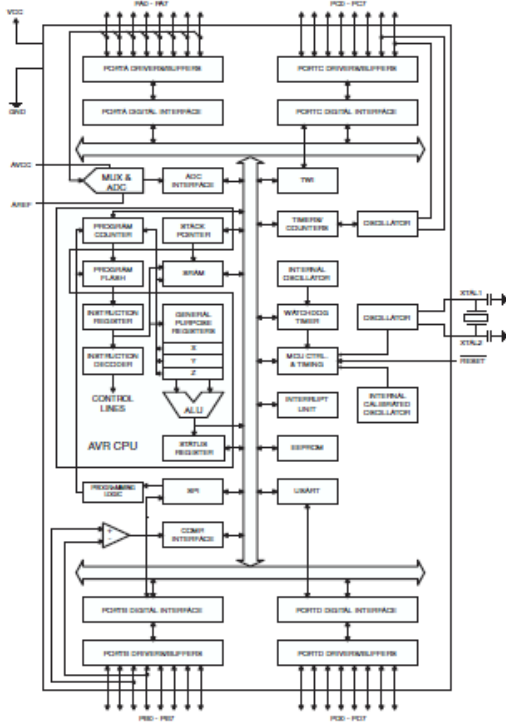
AVR161C AVR161C

Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram





The AVR core combines a rich Instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG Interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire Interface, A/D Converter, SRAM, Timer/Counters, SPI port, and Interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, In-circuit emulators, and evaluation kits.

Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega16 as listed on page 55.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 58.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega16 as listed on page 60.</p>
RESET	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 35. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	<p>Input to the Inverting Oscillator amplifier and input to the internal clock operating circuit.</p>
XTAL2	<p>Output from the Inverting Oscillator amplifier.</p>
AVCC	<p>AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.</p>
AREF	<p>AREF is the analog reference pin for the A/D Converter.</p>

About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

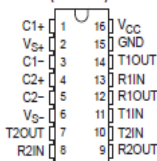
2. Datasheet MAX-232

MAX232, MAX231 DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- \pm 30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202
- Applications
 - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX231 . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept \pm 30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
	0°C to 70°C	PDIP (N)	Tube of 25	MAX232N
SOIC (D)		Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
SOIC (DW)		Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
SOP (NS)	Reel of 2000	MAX232NSR	MAX232	
-40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/omapackage.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production on previous dates not necessarily include testing of all parameters.

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated

1

MAX232, MAX231
DUAL EIA-232 DRIVERS/RECEIVERS

SLLB047L - FEBRUARY 1989 - REVISED MARCH 2004

Function Tables

EACH DRIVER

INPUT T1IN	OUTPUT T1OUT
L	H
H	L

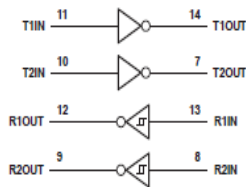
H = high level, L = low level

EACH RECEIVER

INPUT R1IN	OUTPUT R1OUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLL8047L – FEBRUARY 1989 – REVISED MARCH 2004

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)[†]

Input supply voltage range, V_{CC} (see Note 1)	-0.3 V to 6 V
Positive output supply voltage range, V_{S+}	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, V_{S-}	-0.3 V to -15 V
Input voltage range, V_I : Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	± 30 V
Output voltage range, V_O : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Operating virtual junction temperature, T_J	150°C
Storage temperature range, T_{stg}	-65°C to 150°C

[†] 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 under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.

2. Maximum power dissipation is a function of $T_J(\text{max})$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(\text{max}) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.

3. The package thermal impedance is calculated in accordance with JEDEC 51-7.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN, T2IN)	2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)			0.5	V
R1IN, R2IN	Receiver input voltage			± 30	V
T_A	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	-40	85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP [‡]	MAX	UNIT
I_{CC}	Supply current	$V_{CC} = 5.5$ V, All outputs open, $T_A = 25^\circ\text{C}$		8	10 mA

[‡] All typical values are at $V_{CC} = 5$ V and $T_A = 25^\circ\text{C}$.

NOTE 4: Test conditions are C1-C4 = 1 μF at $V_{CC} = 5$ V \pm 0.5 V.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

3

MAX232, MAX2321 DUAL EIA-232 DRIVERS/RECEIVERS

SLL9047L – FEBRUARY 1989 – REVISED MARCH 2004

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH} High-level output voltage	T1OUT, T2OUT $R_L = 3\text{ k}\Omega$ to GND		5	7	V
V_{OL} Low-level output voltage‡	T1OUT, T2OUT $R_L = 3\text{ k}\Omega$ to GND		-7	-5	V
r_O Output resistance	T1OUT, T2OUT $V_{O1} = V_{O2} = 0$, $V_O = \pm 2\text{ V}$	300			Ω
I_{OS}^{\S} Short-circuit output current	T1OUT, T2OUT $V_{CC} = 5.5\text{ V}$, $V_O = 0$		± 10		mA
I_{IS} Short-circuit input current	T1IN, T2IN $V_I = 0$			200	μA

† All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

NOTE 4: Test conditions are C1-C4 = 1 μF at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR Driver slew rate	$R_L = 3\text{ k}\Omega$ to 7 k Ω , See Figure 2			30	V/ μs
SR(t) Driver transition region slew rate	See Figure 3		3		V/ μs
Data rate	One TOUT switching		120		kbit/s

NOTE 4: Test conditions are C1-C4 = 1 μF at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH} High-level output voltage	R1OUT, R2OUT $I_{OH} = -1\text{ mA}$		3.5		V
V_{OL} Low-level output voltage‡	R1OUT, R2OUT $I_{OL} = 3.2\text{ mA}$			0.4	V
V_{IT+} Receiver positive-going input threshold voltage	R1IN, R2IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$		1.7	2.4	V
V_{IT-} Receiver negative-going input threshold voltage	R1IN, R2IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$	0.8	1.2		V
V_{hys} Input hysteresis voltage	R1IN, R2IN $V_{CC} = 5\text{ V}$	0.2	0.5	1	V
R_i Receiver input resistance	R1IN, R2IN $V_{CC} = 5$, $T_A = 25^\circ\text{C}$	3	5	7	k Ω

† All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

NOTE 4: Test conditions are C1-C4 = 1 μF at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ (see Note 4 and Figure 1)

PARAMETER	TYP	UNIT
$t_{C_{LH}(R)}$ Receiver propagation delay time, low- to high-level output	500	ns
$t_{C_{HL}(R)}$ Receiver propagation delay time, high- to low-level output	500	ns

NOTE 4: Test conditions are C1-C4 = 1 μF at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.



POST OFFICE BOX 685333 • DALLAS, TEXAS 75268



DS1307 64 x 8 Serial Real-Time Clock

www.maxim-ic.com

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, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave 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 DIP or SOIC
- Underwriters Laboratory (UL) recognized

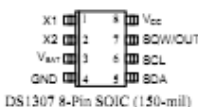
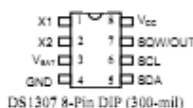
ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional 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 battery supply.

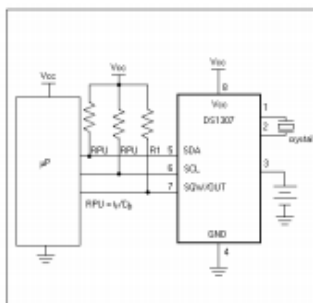
PIN ASSIGNMENT



PIN DESCRIPTION

V _{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V _{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

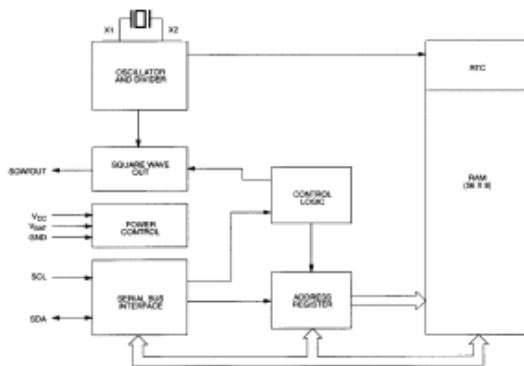
TYPICAL OPERATING CIRCUIT



OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below $1.25 \times V_{BAT}$ the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low-current battery backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than $V_{BAT} + 0.2V$ and recognizes inputs when V_{CC} is greater than $1.25 \times V_{BAT}$. The block diagram in Figure 1 shows the main elements of the serial RTC.

DS1307 BLOCK DIAGRAM Figure 1



2 of 12

SIGNAL DESCRIPTIONS

V_{CC}, GND – DC power is provided to the device on these pins. V_{CC} is the +5V input. When 5V is applied within normal limits, the device is fully accessible and data can be written and read. When a 3V battery is connected to the device and V_{CC} is below 1.25 x V_{BAT}, reads and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below V_{BAT} the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at V_{BAT}.

V_{BAT} – Battery input for any standard 3V lithium cell or other energy source. Battery voltage must be held between 2.0V and 3.5V for proper operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at 25°C. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

See “Conditions of Acceptability” at <http://www.maxim-ic.com/TechSupport/QA/ntrl.htm>.

SCL (Serial Clock Input) – SCL is used to synchronize data movement on the serial interface.

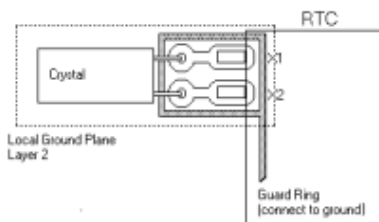
SDA (Serial Data Input/Output) – SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

SQW/OUT (Square Wave/Output Driver) – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pull-up resistor. SQW/OUT will operate with either Vcc or Vbat applied.

X1, X2 – Connections for a standard 32.768kHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5pF.

For more information on crystal selection and crystal layout considerations, please consult Application Note 58, “Crystal Considerations with Dallas Real-Time Clocks.” The DS1307 can also be driven by an external 32.768kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

RECOMMENDED LAYOUT FOR CRYSTAL



6. WIZ110SR Hardware Specification

6.1. Parameters

- Power 5V DC / 180mA
- Dimension 75 x 50 x 17 (L x W x H)
- Temperature Operating Temperature : 0 ~ 80 °C
- Ethernet 10/100 Base-T Ethernet (Auto detection)
- Serial Interface RS-232

6.2. Specification

- MCU 8051 Compliant
- FLASH 62KByte (MCU Internal)
- SRAM 16KByte (MCU Internal)
- EEPROM 2KByte (MCU Internal)

6.3. Board Dimensions and Pin Assignment

6.3.1. Dimensions

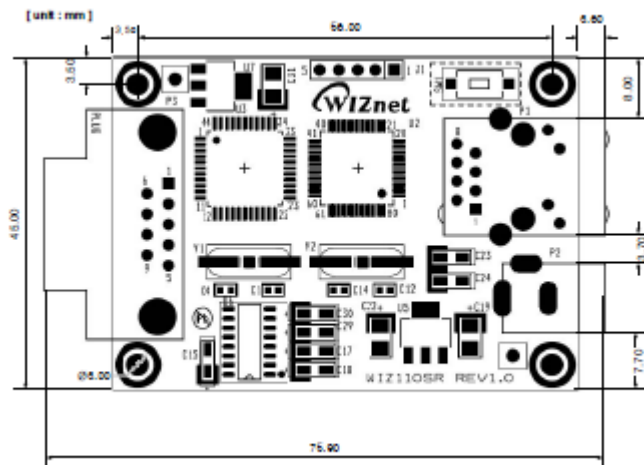


Figure 16. WIZ110SR Dimension

6.3.2. Connector Specification

6.3.2.1. RJ45

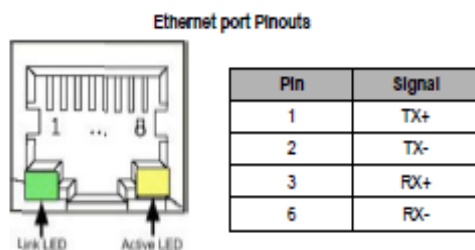
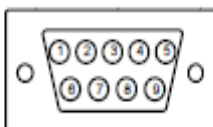


Figure 17. RJ-45 PIN Assignment

6.3.2.2. RS-232

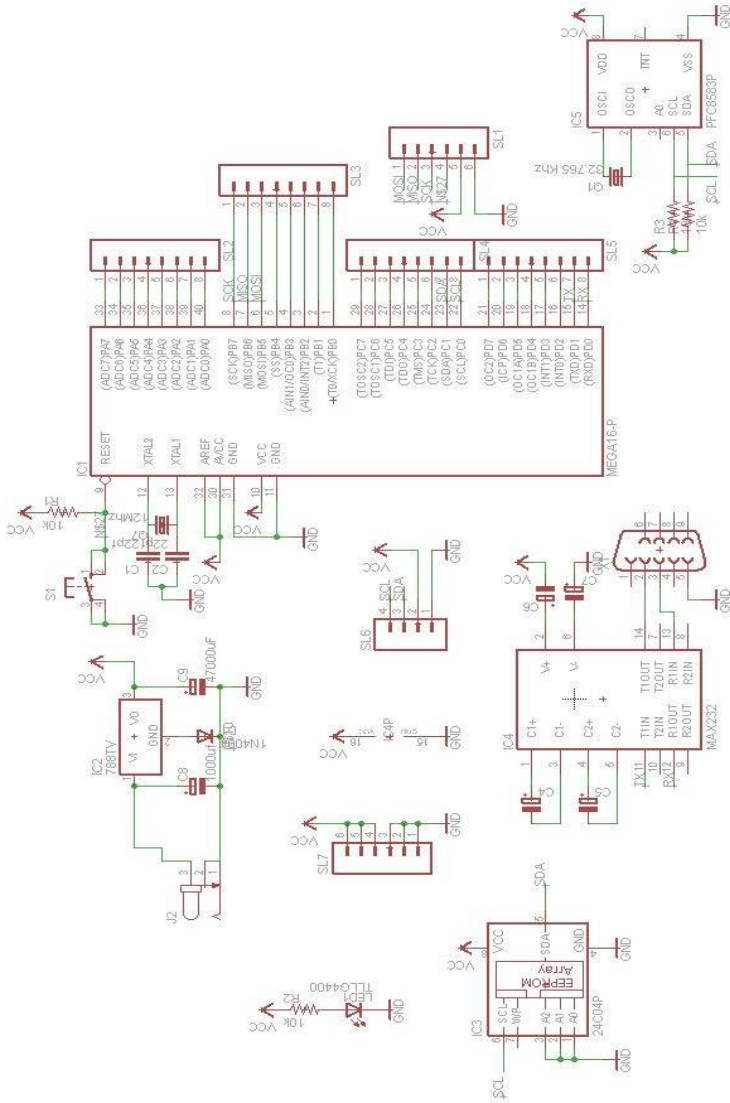


Pin Number	Signal	Description
1	NC	Not Connected
2	RxD	Receive Data
3	TxD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	NC	Not Connected

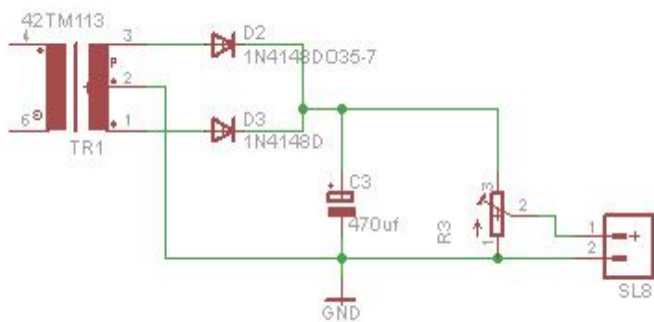
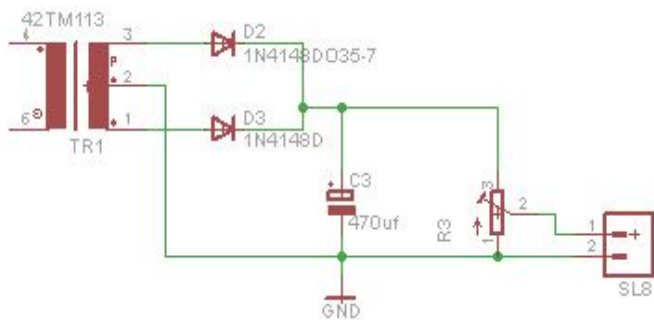
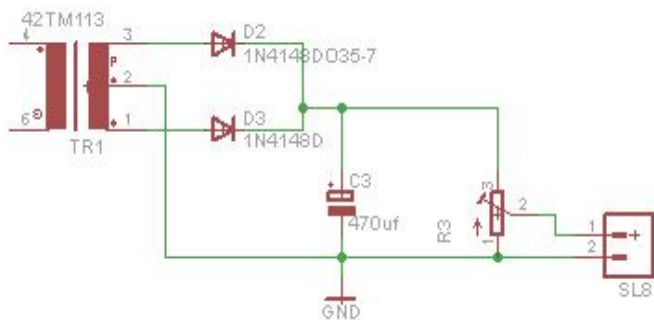
Table 10. RS-232 PIN Assignment

LAMPIRAN 3 SKEMATIK RANGKAIAN

1. Sistem Minimum



2. Rangkaian Sensor Tegangan



DAFTAR RIWAYAT PENULIS

A. KETERANGAN PRIBADI

Nama Lengkap : Regina Yustisia Arifin
Nomor Pokok Mahasiswa : 2212038014
Tempat & Tgl.Lahir : Kediri, 11 April 1995
Jenis kelamin : Perempuan
Alamat Rumah : Jl. Letjen Sutoyo Gg II / 58b Kediri
Alamat di Surabaya : Keputih III / 18a
No. Telpon Rumah : -
No. Telpon Surabaya : -
No. HP : 085655620224
Email : yustisiaregina@gmail.com
Kegemaran/Hobby : menonton film



B. PENDIDIKAN :

S M U : SMAN 1 Kediri Lulus Tahun : 2012
Perguruan Tinggi : D3 Teknik Elektro ITS Lulus Tahun : 2015

C. PENGALAMAN :

Seminar/Simposium/Panitia
Panitia IARC
Ketua Departemen Kesma HIMAD3TEKTRO ITS 2014/2015
Surabaya, Juli 2015
Yang membuat,

(Regina Yustisia Arifin)

Halaman ini sengaja dikosongkan

DAFTAR RIWAYAT PENULIS

A. KETERANGAN PRIBADI

Nama Lengkap : Wahyu Hidayanto
Nomor Pokok Mahasiswa : 2212038021
Tempat & Tgl.Lahir : Madiun, 27 Sept 1994
Jenis kelamin : Laki-laki
Alamat Rumah : Balerejo RT 19 RW 04, Kebonsari,
Madiun
Alamat di Surabaya : Keputih GG 2 no. 26a
No. Telpon Rumah : -
No. Telpon Surabaya : -
No. HP : 085735665878
Email : hidayanto.wahyu@rocketmail.com
Kegemaran/Hobby : menonton film



B. PENDIDIKAN :

S M U : SMAN 1 Geger Lulus Tahun : 2012
Perguruan Tinggi : D3 Teknik Elektro ITS Lulus Tahun : 2015

C. PENGALAMAN :

Seminar/Simposium/Panitia
Panitia IARC

Surabaya, Juli 2015

Yang membuat,

(Wahyu Hidayanto)

Halaman ini sengaja dikosongkan