

ITS
Institut
Teknologi
Sepuluh Nopember

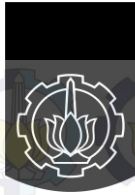
TUGAS AKHIR - TE 145561

**RANCANG BANGUN ALAT PENGHITUNG PENGGUNAAN
ENERGI LISTRIK DAN BIAYA PENGGUNAAN LISTRIK
PADA KAMAR KOST MENGGUNAKAN *ETHERNET*
SEBAGAI MEDIA TRANSMISI DATA**

Medif Zepta Yogassasena
NRP 2213039008
Betty Nur Qamarina
NRP 2213039023

Dosen Pembimbing
Rachmad Setiawan S.T., M.T
Onie Meiyanto, S.Pd

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

DESIGN AND IMPLEMENTATION OF ELECTRICITY AND COST MONITORING AT A BOARDING HOUSE USING ETHERNET AS A DATA TRANSMISSION

Medif Zepta Yogassasena
NRP 2213039008
Betty Nur Qamarina
NRP 2213039023

Advisor
Rachmad Setiawan S.T., M.T
Onie Meiyanto, S.Pd

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini kami menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir kami dengan judul “Rancang Bangun Alat Penghitung Penggunaan Energi Listrik dan Biaya Penggunaan Listrik Pada Kamar Kost Menggunakan *Ethernet* Sebagai Media Transmisi Data” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang kami akui sebagai karya kami sendiri.

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

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

Surabaya, 1 Juni 2016



Medif Zepta Yogassasena
NRP 2213039008



Betty Nur Qamarina
NRP 2213039023



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

**RANCANG BANGUN ALAT PENGHITUNG PENGGUNAAN
ENERGI LISTRIK DAN BIAYA PENGGUNAAN LISTRIK PADA
KAMAR KOST MENGGUNAKAN *ETHERNET* SEBAGAI
MEDIA TRANSMISI DATA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik
Pada**

**Bidang Studi Elektro Industri
Program Studi D3 Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing 1

Rachmad Setiawan, S.T., M.T.
NIP. 19690529 199512 1 001

Dosen Pembimbing 2

Onie Meivanto, S.Pd.
NIP. 19850501 201101 1 008

**SURABAYA
JUNI, 2016**

**LEMBAR PERNYATAAN
PERSETUJUAN PUBLIKASI KARYA ILMIAH
UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, yang bertanda tangan di bawah ini saya :

Nama : Medie Zeptha Yogassasena, Betty Nur Qamafina
Nrp. : 2213039028, 2213039023
Jurusan / Fak. : D3 Teknik Elektro / FTI
Alamat kontak : Perum Wilis Indah 2 Blok H-8 / No. 34 Kediri
a. Email : ungsyam@gmail.com
b. Telp/HP : 085649210482

Menyatakan bahwa semua data yang saya upload di Digital Library ITS merupakan hasil final (revisi terakhir) dari karya ilmiah saya yang sudah disahkan oleh dosen penguji. Apabila dikemudian hari ditemukan ada ketidaksesuaian dengan kenyataan, maka saya bersedia menerima sanksi.

Demi perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalti-Free Right)** kepada Institut Teknologi Sepuluh Nopember Surabaya atas karya ilmiah saya yang berjudul :

Rancang Bangun Alat Penghitung Penggunaan Energi Listrik Dan Biaya Penggunaan Listrik Pada Kamar Kost Menggunakan Etemet Sebagai Media Transmisi Data

Dengan Hak Bebas Royalti Non-Eksklusif ini, Institut Teknologi Sepuluh Nopember Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta. Saya bersedia menanggung secara pribadi, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya Ilmiah saya ini tanpa melibatkan pihak Institut Teknologi Sepuluh Nopember Surabaya.

Demi pernyataan ini saya buat dengan sebenarnya.



NIP. 19950529 199512 1 001

Dibuat di : Surabaya
Pada tanggal : 27 Juni 2016
Yang menyatakan,

Nrp. 2213039023

KETERANGAN :

Tanda tangan pembimbing wajib dibubuhi stempel jurusan.

Form dicetak dan diserahkan di bagian Pengadaan saat mengumpulkan hard copy TA/Tesis/Disertasi.

RANCANG BANGUN ALAT PENGHITUNG PENGGUNAAN ENERGI LISTRIK DAN BIAYA PENGGUNAAN LISTRIK PADA KAMAR KOST MENGGUNAKAN *ETHERNET* SEBAGAI MEDIA TRANSMISI DATA

Nama Mahasiswa :

1. Medif Zepta Yogassasena (2213039008)

2. Betty Nur Qamarina (2213039023)

Dosen Pembimbing :

1. Rachmad Setiawan, S.T., M.T. (19690529 199512 1 001)

2. Onie Meiyanto, S.Pd (19850501 201101 1 008)

ABSTRAK

Penelitian ini membahas masalah penentuan tarif listrik yang dibayarkan oleh setiap penghuni kos, karena kebutuhan energi listrik setiap penghuni kost berbeda-beda. Dengan permasalahan di atas, penentuan biaya listrik kost harus adil agar tidak merugikan salah satu pihak.

Salah satu cara untuk mengatasi masalah di atas, maka dalam Tugas Akhir ini dibuat Alat Penghitung Penggunaan Energi Listrik Dan Biaya Penggunaan Listrik Pada Kamar Kost Menggunakan *Ethernet* Sebagai Media Transmisi Data. Alat ini dapat menentukan konsumsi energi listrik yang digunakan setiap kamar kost sesuai dengan yang digunakan, dengan cara mendapatkan data yang dihasilkan oleh sensor tegangan (*step down transformer*), sensor arus (*Current Transformer*), sensor beda fasa (*Zero Crossing Detector*) dan memprosesnya dengan mikrokontroler. Data penggunaan daya dapat diakses lewat *LabVIEW*.

Hasil simulasi serta implementasi menunjukkan bahwa sistem menghasilkan nilai persentase eror dibawah 5 persen dengan alat ukur yang digunakan, saat kalibrasi sistem.

Kata Kunci : Sensor Arus, Trafo *Step-Down*, *Zero Crossing Detector*, *Ethernet*, Mikrokontroler.



DESIGN AND IMPLEMENTATION OF ELECTRICITY AND COST MONITORING AT A BOARDING HOUSE USING ETHERNET AS A DATA TRANSMISSION

Student Name :

1. Medif Zepta Yogassasena (2213039008)

2. Betty Nur Qamarina (2213039023)

Advisor :

1. Rachmad Setiawan, S.T., M.T. (1969052 9199512 1 001)

2. Onie Meiyanto, S.Pd. (19850501 201101 1 008)

ABSTRACT

This research addresses the problem of determining how much is the electricity cost for the occupant, because electric source needs in every individual is a different even though in the same boarding house. Based on the problem above, the determination of electricity cost must be fair for every occupant so that there would be nobody suffers a financial cost.

The writers suggest one of the solutions to overcome this problem through this final project. In this final project, Design and Implementation of Electricity and Cost Monitoring At A Boarding House Using Ethernet As A Data Transmission. This tool can determine electricity energy consumption that is used in every room in a boarding house. This can be done by getting data which is produced by the sensor voltage (step down transformer), current sensor (Current Transformer), sensor phase difference (Zero Crossing Detector) and process it using microcontroller. The data of the usage of electricity can be accessed through LabVIEW.

The simulation and implementation results show that the system generate value percentage error below 5 percent from the measuring instrument used, when the system is calibrated.

Keywords : Current Sensor, Step Down Transformer, Zero Crossing Detector, Ethernet, Microcontroller.



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Elektro Industri, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN ALAT PENGHITUNG PENGGUNAAN ENERGI LISTRIK DAN BIAYA PENGGUNAAN LISTRIK PADA KAMAR KOST MENGGUNAKAN *ETHERNET* SEBAGAI MEDIA TRANSMISI DATA

Dalam Tugas Akhir ini dirancang alat penghitung penggunaan daya dan biaya penggunaan listrik pada kamar kost menggunakan *Ethernet* sebagai media komunikasinya yang dapat diakses melalui *LabVIEW*.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Rachmad Setiawan, S.T., M.T. dan Bapak Onie Meiyanto, S.Pd. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 1 Juni 2016

Penulis



DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi	3
1.6 Sistematika Laporan.....	4
1.7 Relevansi.....	5
BAB II TEORI DASAR	7
2.1 Tinjauan Pustaka	7
2.2 Daya.....	8
2.2.1 Daya Aktif (Watt).....	8
2.2.2 Daya Reaktif (VAR).....	8
2.2.3 Daya Kompleks (VA).....	9
2.3 Sensor.....	10
2.3.1 Sensor Tegangan	10
2.3.2 Sensor Arus (Transformator Arus)	11
2.3.3 Zero Crossing Detector (ZCD)	12
2.4 Mikrokontroler ATmega16	13
2.5 RS – 232.....	15
2.6 Ethernet Tipe WIZ110SR	16
2.7 LabVIEW.....	17
2.8 LCD 20 x 4.....	18

BAB III PERANCANGAN SISTEM KONTROL	21
3.1 Miniatur Rumah Kost	21
3.2 Blok Fungsional Sistem	23
3.3 Perancangan Perangkat Keras (<i>Hardware</i>)	24
3.3.1 Perancangan <i>Power Supply</i>	24
3.3.2 Sensor Tegangan	25
3.3.3 Sensor Arus	26
3.3.4 Perancangan <i>Zero Crossing Detector</i>	28
3.3.5 Perancangan Rangkaian ATmega16	31
3.3.6 Perancangan Rangkaian MAX232	32
3.4 Perancangan Perangkat Lunak (<i>Software</i>)	33
3.4.1 Perancangan <i>LabVIEW</i>	33
3.4.2 Alur Program Pada Mikrokontroler	35
3.4.3 Perancangan <i>Ethernet</i>	37
BAB IV HASIL SIMULASI DAN IMPLEMENTASI	41
4.1 Pengukuran Rangkaian <i>Power Supply</i>	41
4.2 Pengujian Pin Mikrokontroler	43
4.3 Pengujian Rangkaian Sensor Tegangan	44
4.4 Pengujian Rangkaian Sensor Arus	47
4.5 Pengujian Rangkaian Sensor <i>Zero Crossing Detector</i>	50
4.6 Pengujian Rangkaian Komunikasi	54
4.6.1 Pengujian Rangkaian MAX232	54
4.6.2 <i>Setting IP</i> Rangkaian <i>Ethernet</i> WIZ110SR	56
4.6.3 Pengujian <i>LabVIEW</i>	60
4.7 Pengujian Beban Variasi	61
4.8 Pengujian Keseluruhan	63
BAB V HASIL SIMULASI DAN IMPLEMENTASI	65
5.1 Kesimpulan	65
5.2 Saran	65
DAFTAR PUSTAKA	67
LAMPIRAN A GAMBAR	A-1
LAMPIRAN B PROGRAM	B-1
LAMPIRAN C <i>DATASHEET</i>	C-1
DAFTAR RIWAYAT HIDUP	D-1

DAFTAR GAMBAR

		HALAMAN
Gambar 2.1	Segitiga Daya.....	8
Gambar 2.2	Trafo <i>Step-Down</i>	11
Gambar 2.3	<i>Current Transformator</i> 5A/2,5mA12	12
Gambar 2.4	<i>Input dan Output Zero Crossing Detector</i>13	13
Gambar 2.5	Konfigurasi Mikrokontroler ATmega16.....14	14
Gambar 2.6	Konfigurasi RS-23215	15
Gambar 2.7	<i>Ethernet Tipe W1Z110SR</i>17	17
Gambar 2.8	Tampilan <i>Front panel</i> dan <i>Block Diagram Pada LabVIEW</i>18	18
Gambar 2.9	LCD 20 x 419	19
Gambar 3.1	Sketsa Miniatur Rumah Kost21	21
Gambar 3.2	Sketsa Lantai 1.....22	22
Gambar 3.3	Sketsa Lantai 2.....23	23
Gambar 3.4	Blok Fungsional.....23	23
Gambar 3.5	Rangkaian <i>Power Supply</i>24	24
Gambar 3.6	Rangkaian Simulasi Sensor Tegangan25	25
Gambar 3.7	Sensor Arus <i>Simple Phase AC</i> 5A26	26
Gambar 3.8	Grafik Sensor Arus <i>Simple Phase AC</i> 5A27	27
Gambar 3.9	Rangkaian Simulasi Sensor Arus.....28	28
Gambar 3.10	Rangkaian <i>Zero Crossing Detector</i> Percobaan 129	29
Gambar 3.11	<i>Output Sensor Tegangan</i>29	29
Gambar 3.12	<i>Output XOR</i>29	29
Gambar 3.13	<i>Output Sensor Arus</i>30	30
Gambar 3.14	<i>Output XOR Saat Tersambung Ground</i>30	30
Gambar 3.15	Rangkaian <i>Zero Crossing Detector</i> Percobaan 231	31
Gambar 3.16	Perancangan Rangkaian Minimum ATmega1631	31
Gambar 3.17	Rangkaian MAX 23233	33
Gambar 3.18	Tampilan Blok Diagram Pada <i>LabVIEW</i>34	34
Gambar 3.19	Tampilan <i>Front panel</i> Pada <i>LabVIEW</i>35	35
Gambar 3.20	<i>Flow Chart</i> Sistem36	36
Gambar 3.21	Tampilan <i>Software W1Z110SR Configuration Tool</i>37	37
Gambar 4.1	Pengukuran Tanpa Beban.....42	42
Gambar 4.2	Pengukuran dengan Menggunakan Beban42	42
Gambar 4.3	Program Uji Coba Mikrokontroler.....42	42
Gambar 4.4	Grafik ADC Terhadap Tegangan Kamar 146	46

Gambar 4.5	Grafik ADC terhadap Tegangan Pada Kamar 2	47
Gambar 4.6	Grafik Arus Terhadap ADC Pada Kamar 1	49
Gambar 4.7	Grafik Arus Terhadap ADC Pada Kamar 2	50
Gambar 4.8	<i>Output</i> LM 741 dari Sensor Tegangan Pada Kamar 1	51
Gambar 4.9	<i>Output</i> LM 741 dari Sensor Arus Pada Kamar 1	51
Gambar 4.10	<i>Output</i> XOR Pada Kamar 1	51
Gambar 4.11	<i>Output</i> LM 741 dari Sensor Tegangan Pada Kamar 2	52
Gambar 4.12	<i>Output</i> LM 741 dari Sensor Arus Pada Kamar 2	52
Gambar 4.13	<i>Output</i> XOR Pada Kamar 2	53
Gambar 4.14	Skema Rangkaian Komunikasi	54
Gambar 4.15	Tampilan <i>Hyper Terminal</i>	55
Gambar 4.16	Tampilan <i>Testing MAX232</i>	55
Gambar 4.17	Tampilan <i>Port Setting</i>	56
Gambar 4.18	Tampilan <i>WIZ10XSR Configuration Tool</i>	57
Gambar 4.19	Pengisian <i>IP Address Ethernet WIZ110SR</i>	58
Gambar 4.20	Pengiriman Data untuk Memanggil Router	58
Gambar 4.21	Tampilan <i>Software RealTerm</i>	59
Gambar 4.22	Penerimaan Data Pada <i>RealTerm</i>	59
Gambar 4.23	Tampilan LabVIEW Dengan Tarif 2000 per kWh	60
Gambar 4.24	Tampilan LabVIEW Dengan Tarif 1264.86 per kWh	60
Gambar 4.25	Tampilan LabVIEW Tanpa Beban	63
Gambar 4.26	Tampilan <i>LabVIEW</i> Pada Kamar 1	63
Gambar 4.27	Tampilan <i>LabVIEW</i> Pada Kamar 2	64

DAFTAR TABEL

		HALAMAN
Tabel 2.1	Keterangan Konfigurasi Pada RS-232	15
Tabel 2.2	Spesifikasi Modul <i>Ethernet</i> W1Z110SR.....	16
Tabel 2.3	Konfigurasi Pin LCD	19
Tabel 3.1	Data Sensor Arus <i>Simple Phase</i> AC 5A.....	27
Tabel 4.1	Hasil Pengukuran <i>Power Supply</i> 5 Volt, -5 Volt Tanpa Beban	42
Tabel 4.2	Hasil Pengukuran <i>Power Supply</i> 5 Volt dengan Beban ..	42
Tabel 4.3	Hasil Pengukuran Tegangan pada Setiap Pin Mikrokontroler	44
Tabel 4.4	Pengambilan Data Sensor Tegangan Kamar 1	45
Tabel 4.5	Pengambilan Data Sensor Tegangan Kamar 2	45
Tabel 4.6	Pengambilan Data Sensor Arus Kamar 1	48
Tabel 4.7	Pengambilan Data Sensor Arus Kamar 2	48
Tabel 4.8	Data Pengambilan <i>CosPhi</i>	53
Tabel 4.9	Data Penggunaan Daya Kamar 1	61
Tabel 4.10	Data Penggunaan Daya Kamar 2	62



BAB I

PENDAHULUAN

1.1 Latar Belakang

Rumah kost merupakan tempat tinggal sementara untuk mahasiswa, pelajar, pekerja, atau siapapun yang beraktifitas jauh dari rumah atau sedang merantau. Dan pada saat ini sering dijumpai atau bahkan pernah mengalami permasalahan pembayaran tagihan listrik pada rumah kost yang kadang tidak sesuai dengan pemakaian sebenarnya, sehingga hal itu dapat merugikan penyewa kamar kost. Pada saat penyewa kost tidak menempati kamar kost untuk beberapa hari atau bahkan beberapa minggu tapi pada akhirnya biaya listrik yang diberikan pada penyewa kost masih tetap sama seperti hari-hari biasa saat penyewa kost masih tinggal di kost.

Mungkin hal tersebut kurang disadari oleh para penyewa kost, hal ini dapat terjadi dikarenakan biaya listrik pada tiap kamar kost hanya dikhususkan pada peralatan-peralatan elektronik yang digunakan pada kamar tersebut, seperti kipas angin, komputer, setrika dll. Sehingga penyewa kamar kost akan merasa dirugikan apabila peralatan-peralatan elektronik tersebut tidak atau jarang digunakan. Maka dari itu, perlu direalisasikan suatu alat yang dapat menentukan tarif listrik sesuai yang digunakan oleh penghuni kamar. Hal inilah yang menjadi dasar dalam pembuatan Tugas Akhir berupa alat yang dapat menentukan tarif listrik sesuai yang digunakan penghuni kamar yang kemudian bisa di akses oleh penghuni kamar melalui *LabVIEW*. Tujuannya adalah agar mempermudah dalam memantau penggunaan energi listrik.

1.2 Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini adalah sebagai berikut :

- Tidak sesuaiinya pembagian tarif listrik yang digunakan oleh setiap penghuni kost.
- Belum ada *monitoring* penggunaan energi listrik setiap penghuni kost
- Privasi tentang pemakaian listrik oleh penghuni kamar kurang terjaga.
- Instrumentasi untuk mengukur tegangan jala-jala.

- Instrumentasi untuk mengukur arus AC karena beban.
- Instrumentasi untuk mengukur beda fase antara tegangan dan arus AC untuk menentukan faktor daya.
- Cara penggunaan modul *Ethernet*.
- Cara komunikasi data internet dengan *Ethernet* di mikrokontroler.
- Cara menentukan tarif dan melakukan perubahan tarif jika terjadi perubahan tarif dari pemerintah.

1.3 Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari Tugas Akhir ini adalah :

- Penggunaan daya yang digunakan hanya bisa dimonitoring melalui *LabVIEW* dan ditampilkan di LCD.
- Media komunikasi yang digunakan adalah *Ethernet*..
- Pengontrolan menggunakan ATmega16.
- Sensor yang digunakan sebagai pembaca penggunaan daya adalah sensor arus, sensor tegangan yang menggunakan pembagi tegangan dan sensor *Zero Crossing Detector* yang digunakan untuk mendeteksi beda fasa.

1.4 Tujuan

Penelitian ini bertujuan sebagai berikut :

- Merancang dan membuat suatu alat yang dapat mengetahui berapa biaya listrik yang harus dibayarkan penghuni kost sesuai dengan pemakaiannya
- Mempermudah untuk memonitoring penggunaan energi listrik setiap penghuni kost
- Menjaga privasi penghuni kost tentang pemakaian energi listrik tanpa diketahui orang lain.
- Merancang dan membuat instrumentasi untuk mengukur tegangan jala-jala.
- Merancang dan membuat instrumentasi untuk mengukur arus AC karena beban.
- Merancang dan membuat instrumentasi untuk mengukur beda fase antara tegangan dan arus AC untuk menentukan faktor daya.
- Memberikan langkah – langkah dalam penggunaan *Ethernet*.
- Merancang komunikasi data internet dengan *Ethernet* di mikrokontroler.

- Merancang dan membuat program perhitungan kWh di dalam mikrokontroler.
- Mempermudah pengguna dalam menentukan tarif dan perubahan tarif jika terjadi perubahan tarif dari pemerintah.

1.5 Metodologi

Dalam pelaksanaan Tugas Akhir berupa Rancang Bangun Alat Penghitung Penggunaan Energi Listrik dan Biaya Penggunaan Listrik Pada Kamar Kost Menggunakan *Ethernet* Sebagai Media Transmisi Data, ada beberapa kegiatan yang dapat diuraikan sebagai berikut :

- Studi Pustaka
Studi pustaka yang dilakukan dalam hal ini adalah dengan mencari literatur di ruang baca, perpustakaan atau di internet mengenai penggunaan *Ethernet*, mengenai sensor arus, sensor tegangan, dan sensor beda fasa serta AVR ATmega. Untuk survei data awal yang dilakukan dalam hal ini adalah dengan melakukan survei penentuan tarif dasar listrik pada kost.
- Studi Sistem
Dalam pengambilan data digunakan data tentang tarif listrik pada rumah, dan mengamati sistem dari kWh meter.
- Perancangan Sistem
Setelah mengetahui gambaran umum sistem, maka dilanjutkan dengan perancangan sistem. Perancangan dimulai dengan pembacaan tegangan jala-jala. Karena kemampuan mikrokontroler yang hanya bisa membaca tegangan DC maksimum 5 volt, maka dibutuhkan Trafo *step down* dan rangkaian *peak detector*. Perancangan berikutnya adalah membaca arus beban. Karena mikrokontroler hanya bisa membaca tegangan dan maksimum 5 volt, maka dibutuhkan trafo arus dan rangkaian *peak detector*. Karena daya listrik AC ada, $V \times I \times \cos \phi$ maka harus dibuat detektor beda fasa. Data pemakaian daya di-*monitoring* dari *LabVIEW* menggunakan RS 232 yang disambung dengan kabel RJ45 dan dapat diakses melalui PC.

- **Pembuatan *Hardware***

Tahapan ini dilakukan setelah mendapat informasi dari referensi di atas. Pembuatan dimulai dengan pembuatan rangkaian sensor tegangan yang menggunakan *Trafo step down*, pembuatan rangkaian sensor arus yang menggunakan *Current Transformer* dan pembuatan sensor beda fasa menggunakan rangkaian *Zero Crossing Detector*.

- **Pembuatan *Software***

Pembuatan *software* dilakukan setelah perancangan dan pembuatan alat selesai dilakukan, pembuatan *software* ini dilakukan pada mikrokontroler (AVR ATmega) dan perancangan sistem *Ethernet* yang menggunakan RS 232 yang tersambung dengan TCP/IP agar bisa tersambung dengan internet.

- **Uji Coba dan Analisis Data**

Uji coba dan analisis data dilakukan dengan cara mengaplikasikan langsung alat ini pada instalasi listrik dan mengirimkan data melalui *Ethernet* serta menganalisa ketepatan hasil alat dengan hasil sebenarnya.

- **Evaluasi dan Penyempurnaan Alat**

Pada tahap ini data yang diperoleh pada saat pengujian alat awal digunakan sebagai acuan untuk menganalisa kinerja sistem menggunakan perhitungan dan interpretasi data sebagai tindak lanjut dan penyempurnaan alat, sehingga alat tersebut dapat mengolah data yang benar sesuai dengan rencana awal.

- **Penyusunan Laporan**

Penyusunan laporan dilakukan setelah beberapa data pengujian alat penentu tarif listrik pada kost yang didapat sudah mencukupi.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, teori tentang sensor arus, sensor tegangan, *Zero Crossing Detector*, teori tentang ATmega16, RS-232, *Ethernet* Tipe W1Z110SR, *LabVIEW*, dan LCD 16x4.

Bab III Perancangan Sistem

Bab ini membahas tentang perencanaan dan pembuatan perangkat keras (*hardware*) yang terdiri dari perancangan elektronik dan perancangan mekanik serta pembuatan dan perancangan perangkat lunak (*software*).

Bab IV Simulasi, Implementasi dan Analisis Sistem

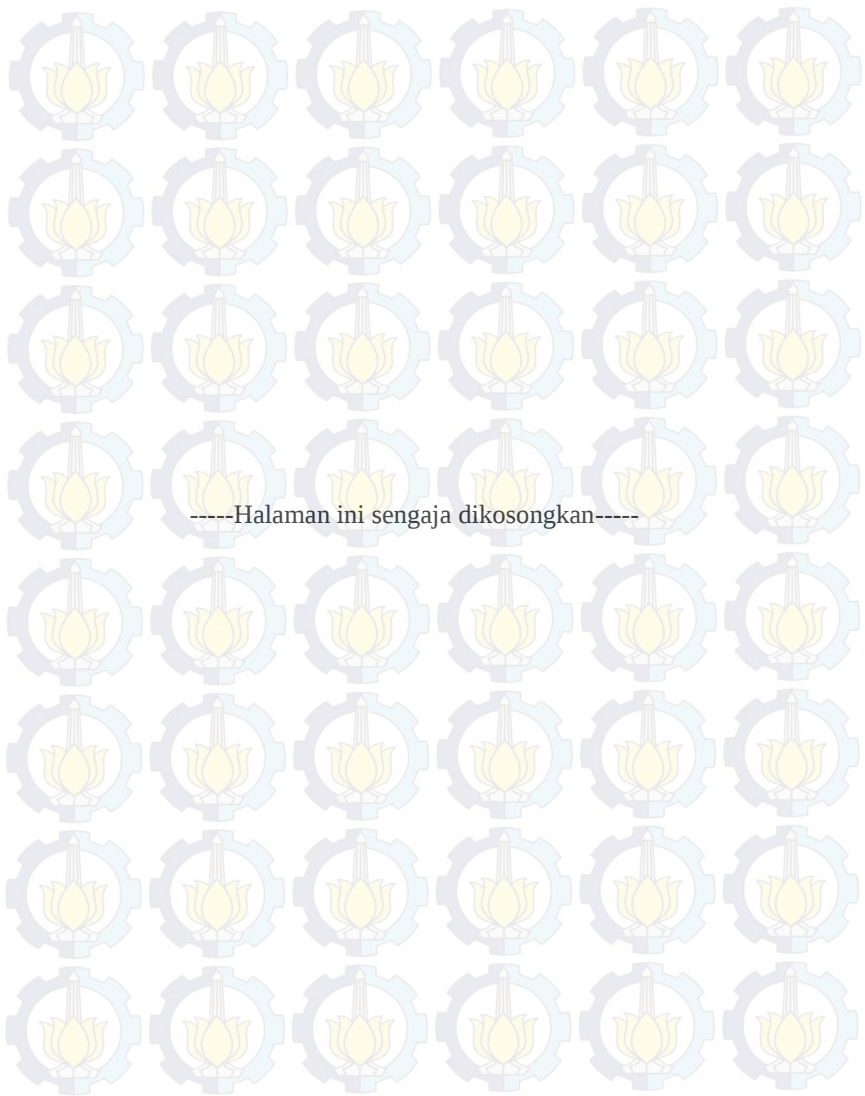
Bab ini memuat tentang hasil pengujian alat yang terdiri dari pengujian perangkat keras dan juga perangkat lunak dengan pengukuran.

Bab V Penutup

Bab ini berisi tentang kesimpulan dari Tugas Akhir ini dan saran untuk pengembangan alat ini lebih lanjut.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadi referensi perencanaan desain *monitoring* daya pada kamar kost, agar penentuan tarif daftar listrik mempunyai metode yang adil dan modern.



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

BAB II

TEORI DASAR

2.1 Tinjauan Pustaka

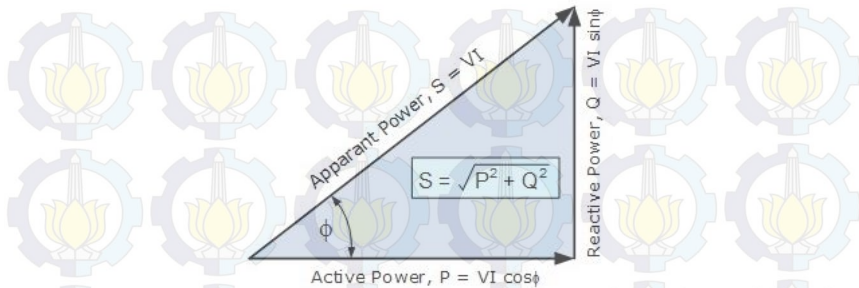
Ada beberapa metode yang pernah diusulkan untuk menyelesaikan masalah *monitoring* penggunaan daya, metode pertama memfokuskan *monitoring* penggunaan daya di kamar kost dengan menggunakan intranet sebagai media transmisi data, dan metode kedua memfokuskan *monitoring* penggunaan daya di gedung bertingkat.

Pada metode pertama menggunakan sensor arus Alegro ACS712 untuk mengetahui arus yang mengalir dari sumber diperlukan suatu rangkaian yang dapat mengubah besaran arus menuju ke sebuah peralatan elektronika dari sebuah sumber tegangan. Untuk sensor tegangan digunakan Trafo *step down* 500 mA, yang dimaksudkan agar keluaran dari trafo stabil dengan keluaran 3,5 Volt sehingga keluaran sinyal tersebut dapat dibaca oleh ADC dengan tegangan referensi 5 Volt [1].

Pada metode kedua menggunakan sensor arus dari trafo bekas pemakaian radio yang difungsikan sebagai trafo arus. Caranya adalah melilitkan kawat email yang dialiri listrik menuju beban di salah satu sisi trafo yang dialiri listrik menuju beban maka trafo CT tersebut akan terinduksi oleh arus yang mengalir pada beban sehingga menimbulkan tegangan pada trafo CT sisi sekunder tersebut. Besarnya tegangan yang diakibatkan arus yang mengalir pada sisi primer lewat impedansinya menimbulkan tegangan yang proposional disisi sekunder sebagai representasi arus, tetapi hal tersebut mengakibatkan nilai induktansi yang tinggi. Sensor tegangannya memakai pembagi tegangan (*voltage divider*) [2].

Pada Tugas Akhir ini dilakukan perancangan alat yang menggabungkan dua metode diatas yang terdiri dari perancangan sensor arus menggunakan *current transformer* 5A/2,5mA, yang diharapkan *ripple* dari *output* CT tidak mempunyai nilai induktansi yang tinggi, dan sensor tegangan menggunakan Trafo *step down* 350 mA, yang dimaksudkan agar keluarannya sebesar 6 Volt, setelah itu rangkaian sensor tegangan masuk ke *peak detector* untuk mengetahui nilai puncak tegangan yang dihasilkan. Hasil yang diharapkan adalah data yang diambil lebih akurat dan lebih mudah dalam mengaksesnya yaitu menggunakan *LabVIEW*.

2.2 Daya



Gambar 2.1 Segitiga Daya

Daya adalah energi yang dikeluarkan untuk melakukan usaha. Dalam sistem tenaga listrik, daya merupakan jumlah energi yang digunakan untuk melakukan kerja atau usaha. Dibawah ini merupakan segitiga daya yang menghubungkan daya aktif, daya reaktif, dan daya kompleks [3].

2.2.1 Daya Aktif (Watt)

Secara sederhana, daya aktif adalah daya yang dibutuhkan oleh beban resistif. Daya aktif menunjukkan adanya aliran energi listrik dari pembangkit listrik ke jaringan beban untuk dapat dikonversikan menjadi energi lain. Sebagai contoh, daya aktif yang digunakan untuk menyalakan kompor listrik. Energi listrik yang mengalir dari jaringan dan masuk ke kompor listrik, dikonversikan menjadi energi panas oleh elemen pemanas kompor tersebut.

Penghitungan daya aktif ini pada arus DC dengan arus AC terdapat sedikit perbedaan. Daya listrik pada arus listrik DC, dirumuskan sebagai perkalian arus listrik dengan tegangan.

$$P = V \times I \dots\dots\dots (2.1)$$

Namun pada listrik AC perhitungan daya menjadi sedikit berbeda karena melibatkan faktor daya ($\cos \varphi$), rumus seperti pada Gambar 2.1.

$$P = V \times I \times \cos \varphi \dots\dots\dots (2.2)$$

2.2.2 Daya Reaktif (VAR)

Daya reaktif adalah daya yang menunjukkan adanya pergeseran grafik sinusoidal arus dan tegangan listrik AC akibat adanya beban reaktif. Daya reaktif memiliki fungsi yang sama dengan faktor daya atau

juga bilangan $\cos \phi$. Daya reaktif ataupun faktor daya akan memiliki nilai ($\neq 0$) jika terjadi pergeseran grafik sinusoidal tegangan ataupun arus listrik AC, yakni pada saat beban listrik AC bersifat induktif ataupun kapasitif. Sedangkan jika beban listrik AC bersifat murni resistif, maka nilai dari daya reaktif akan nol ($=0$).

Pengendalian daya reaktif pada sistem jaringan distribusi listrik AC sangat penting untuk diperhatikan. Hal ini tidak lepas dari pengaruh beban reaktif terhadap kondisi jaringan listrik AC. Beban kapasitif yang bersifat menyimpan tegangan sementara, cenderung mengakibatkan nilai tegangan jaringan menjadi lebih tinggi daripada yang seharusnya. Sedangkan beban induktif yang bersifat menyerap arus listrik, cenderung membuat tegangan listrik jaringan turun. Berubah-ubahnya tegangan listrik jaringan tersebut sangat mengganggu proses distribusi energi listrik dari pembangkit ke konsumen. Perubahan tegangan jaringan berkaitan langsung dengan kerugian-kerugian distribusi listrik seperti kerugian panas dan emisi elektromagnetik yang terbentuk sepanjang jaringan distribusi. Semakin jauh nilai tegangan jaringan dari angka yang seharusnya, akan semakin besar kerugian distribusi listriknya dan akan semakin mengganggu proses distribusi daya nyata listrik. Di sinilah peran kontrol daya reaktif jaringan listrik sangat perlu diperhatikan. Rumus dari daya kompleks, seperti terlihat dari segitiga daya pada Gambar 2.1 :

$$Q = V \times I \times \sin \varphi \dots \dots \dots (2.3)$$

2.2.3 Daya Kompleks (VA)

Daya kompleks adalah daya yang dihasilkan dari perkalian tegangan dan arus listrik. Daya kompleks merupakan daya yang diberikan oleh PLN kepada konsumen. Satuan daya kompleks adalah VA (Volt.Ampere).

Beban yang bersifat daya kompleks adalah beban yang bersifat resistif (R), contoh : lampu pijar, setrika listrik, kompor listrik dan lain sebagainya. Peralatan listrik atau beban pada rangkaian listrik yang bersifat resistif tidak dapat dihemat karena tegangan dan arus listrik se fasa, perbedaan sudut fasa adalah 0° dan memiliki nilai faktor daya adalah 1. Daya kompleks didefinisikan sebagai besaran daya yang bagian realnya daya nyata dan bagian imajiner daya reaktif yang dapat dituliskan sebagai berikut :

$$\begin{aligned} S &= P + jQ \\ S &= V \times I \dots \dots \dots (2.4) \end{aligned}$$

Setiap peralatan listrik di rumah sebenarnya hanya mencantumkan nilai daya listrik dalam Watt, yang merupakan daya aktif. contohnya mesin *jetpump* 150Watt, lampu TL 20Watt, AC 300Watt dan lain-lain. Bila semua peralatan listrik tersebut dipakai, maka total maksimum daya yang mampu disediakan hanya 1056Watt (bila rumah tersebut berlangganan listrik 1300VA).

Untuk pelanggan perumahan, hanya penggunaan daya aktif dalam satuan watt yang dihitung oleh PLN. Karena itu alat pengukurnya disebut kWh-meter (Kilo-Watt Hour meter). Satu kWh adalah pemanfaatan energi listrik sebesar 1000 Watt dalam waktu 1 jam. Besarnya daya reaktif tidak dihitung karena faktor daya untuk listrik perumahan masih ditoleransi dalam angka 0.8. Berbeda dengan listrik industri dimana terpasang kVARh-meter (Kilo-VAR Hour meter) untuk menghitung besarnya pemakaian daya reaktif, dimana jika penggunaannya melebihi batas maka akan kena pinalti oleh PLN.

Jadi satuan Watt lebih digunakan untuk menghitung besarnya penggunaan daya listrik pada peralatan dan satuan VA digunakan untuk menghitung kapasitas terpasang instalasi listrik, mulai dari MCB dan penghantarnya.

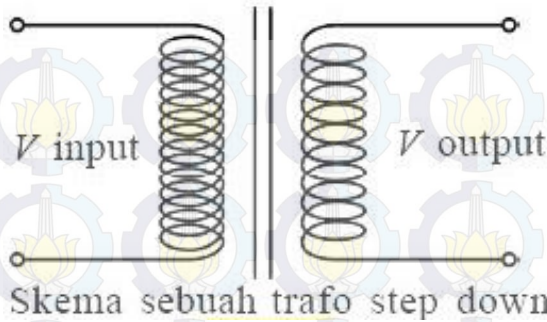
Pada Tugas Akhir ini adalah untuk menghitung daya aktif seperti rumus 2.1 yang digunakan pelanggan dengan menggunakan tiga sensor, yaitu sensor tegangan untuk mendeteksi besar “V”, sensor arus untuk mendeteksi besar “I”, dan sensor *Zero Crossing Detector* untuk mendeteksi “ $\cos \varphi$ ”

2.3 Sensor

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang dirubah menjadi besaran listrik disebut *Transduser*. Pada proyek akhir ini, sensor yang digunakan adalah sensor tegangan dan sensor arus, sensor beda fasa (*Zero Crossing Detector*).

2.3.1 Sensor Tegangan

Transformator yang menghasilkan tegangan sekunder kurang dari tegangan primer disebut *Trafo step down*. Jumlah tegangan yang diturunkan ditentukan oleh rasio belitan antara belitan primer dan sekunder. Rasio belitan selalu kurang dari satu [4].



Gambar 2.2 Trafo Step-Down

Rumus dari Trafo Step-Down, sebagai berikut :

$$\frac{V_s}{V_p} = \frac{N_s}{N_p} = \frac{I_p}{I_s} \dots\dots\dots (2.5)$$

Jadi, belitan primer selalu lebih besar dari lilitan sekunder ($N_p > N_s$).
Tegangan primer selalu lebih besar dari tegangan sekunder ($V_p > V_s$).
Kuat arus primer selalu lebih kecil dari kuat arus sekunder ($I_p < I_s$).

2.3.2 Sensor Arus (Transformator Arus)

Dalam sistem kelistrikan Trafo arus (CT) / *Current Transformer* di gunakan untuk pengukuran arus listrik. *Current Transformer* hampir sama dengan VT trafo tegangan atau sering di sebut dengan (PT) *Potential Transformer*, keduanya di kenal dengan *instrument transformer*. Di saat Arus terlalu tinggi dalam jaringan maka di perlukan CT untuk *converter* pembacaan pada alat ukur jadi yang digunakan progresif arus imbas dari hantaran dari sebuah rangkaian listrik bolak balik atau AC. Sebuah trafo arus menghasilkan konversi arus yang akurat untuk pembacaan alat ukur atau sensor *safety device* [5].

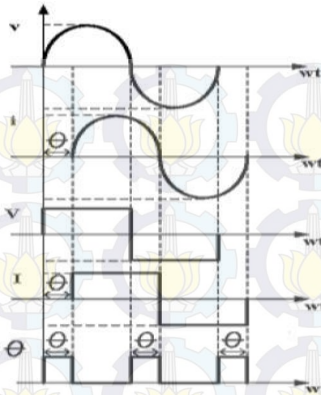


Gambar 2.3 *Current Transformer 5A/2,5mA*

2.3.3 Zero Crossing Detector (ZCD)

Rangkaian ini berfungsi untuk mendeteksi perbedaan sudut fasa yang mengalir ke beban. *Detektor Fasa* dibuat menggunakan komparator dan gerbang logika XOR. Komparator digunakan untuk mendapatkan informasi saat nilai tegangan dan nilai arus tepat melewati titik nol. Gerbang logika XOR digunakan untuk mengetahui nilai beda sudut fasa. Nilai perbedaan sudut fasa didapat dengan menghitung selang waktu antara tegangan naik dan tegangan turun pada keluaran gerbang logika XOR.

Sinyal masukan pada komparator berupa sinyal sinusoidal yang berasal dari sensor arus dan sensor tegangan. Oleh komparator diubah menjadi sinyal kotak sebagai masukan gerbang logika XOR. Keluaran dari XOR berupa pulsa beda fasa antara arus dan tegangan [6].



Gambar 2.4 *Input dan Output Zero Crossing Detector*

Teori rumus mencari $\cos \varphi$.

$$\cos \varphi = \frac{t_{on}}{t_{on}} \times 360^\circ \quad (2.6)$$

$$\cos \varphi = \frac{t_{on}}{\frac{1}{2} T} \times 360^\circ \quad (2.7)$$

Rumus mencari nilai $\cos \varphi$ berdasarkan *output Zero Crossing Detector*.

$$\cos \varphi = \frac{t_c \times 100 \mu s}{1000 \mu s} \times 180^\circ \quad (2.8)$$

Keterangan :

T_c = Total Counter

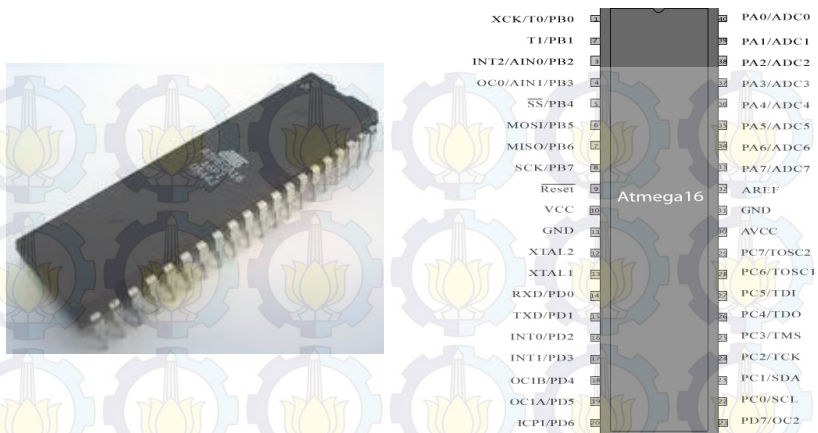
$\cos \varphi$ = sudut $\cos \varphi$

T = periode (detik)

Maka setiap 1 counter adalah 1.8°

2.4 Mikrokontroler ATmega16

ATmega16 adalah mikrokontroler CMOS 8 bit daya rendah berbasis arsitektur RISC. Instruksi dikerjakan pada satu siklus *clock*, ATmega16 mempunyai *throughput* mendekati 1 MIPS per MHz, hal ini membuat ATmega16 dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah [7].



Gambar 2.5 Konfigurasi Mikrokontroler ATmega16

Mikrokontroler ATmega16 memiliki beberapa fitur atau spesifikasi yang menjadikannya sebuah solusi pengendali yang efektif untuk berbagai keperluan. Fitur-fitur tersebut antara lain:

1. Saluran I/O ada 32 buah, yaitu Port A, Port B, Port C, dan PortD.
2. ADC (Analog to Digital Converter) 10 bit sebanyak 8 channel.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri dari 32 register.
5. 131 intruksi andal yang umumnya hanya membutuhkan 1 siklus *clock*.
6. Watchdog *Timer* dengan oscilator internal.
7. Dua buah *Timer/Counter* 8 bit.
8. Satu buah *Timer/Counter* 16 bit.
9. Tagangan operasi 2.7 V - 5.5 V pada ATmega16.
10. *Internal SRAM* sebesar 1KB.
11. *Memory Flash* sebesar 16KB dengan kemampuan *Read While Write*.
12. Unit interupsi internal dan eksternal.
13. Port antarmuka SPI.
14. *EEPROM* sebesar 512 byte dapat diprogram saat operasi.
15. Antar muka komparator analog.
16. 4 channel PWM.
17. 32x8 general purpose register.

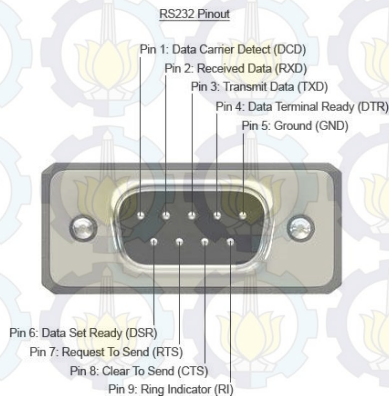
18. Hampir mencapai 16 MIPS pada Kristal 16 MHz.
19. Port USART *programmable* untuk komunikasi serial.

2.5 RS – 232

RS 232 adalah *standard* komunikasi serial yang digunakan untuk koneksi periperal ke periperal. Biasa juga disebut dengan jalur I/O (*input / output*). Contoh yang paling sering ditemui adalah koneksi antara komputer dengan modem, atau komputer dengan mouse bahkan bisa juga antara komputer dengan komputer, semua biasanya

Tabel 2.1 Keterangan Konfigurasi Pada RS-232

Pin DB25	Pin DB 9	Singkatan	Keterangan
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indikator



Gambar 2.6 Konfigurasi RS-232

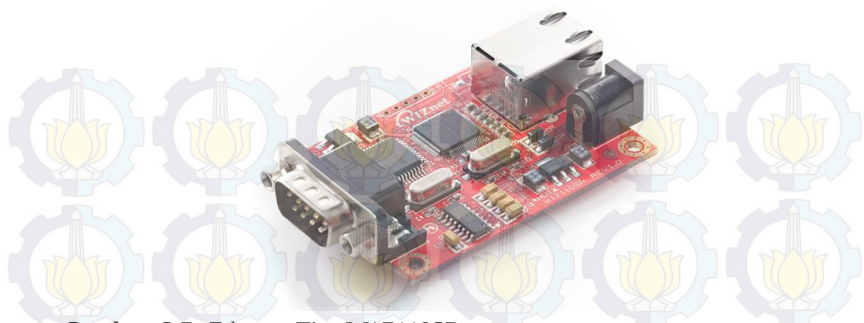
dihubungkan lewat jalur port serial RS232. Standar ini menggunakan beberapa piranti dalam implementasinya. Paling umum yang dipakai adalah plug / konektor DB9 atau DB25 [8].

2.6 Ethernet Tipe WIZ110SR

Gateway modul yang mengkonversi protokol RS-232 ke TCP / IP protokol. Modul ini mengukur, mengelola dan mengendalikan perangkat melalui jaringan berbasis pada *Ethernet* dan TCP / IP dengan menghubungkan ke peralatan yang ada dengan serial RS-232 *interface*. Dengan kata lain, WIZ110SR merupakan sebuah protokol konverter yang mentransmisikan data yang dikirim dengan serial peralatan sebagai tipe data TCP / IP dan mengkonversi kembali data TCP / IP yang diterima melalui jaringan ke data serial untuk mengirimkan kembali ke peralatan. Dibawah ini merupakan spesifikasi modul *Ethernet* WIZ110SR:[9]

Tabel 2.2 Spesifikasi Modul *Ethernet* WIZ110SR

Item	Gambaran
MCU	8051 Compliant (Internal 62K Flash, 16K SRAM, 2K EEPROM)
TCP/IP	W5100 (<i>Ethernet</i> PHY Embedded)
Network Interface	10/100 Mbps auto-sensing RJ-45 Connector
Serial Interface	RS 232
Serial Signal	TXD, RXD, RTS, CTS, GND
Serial Parameters	Parity : None, Even, Odd
	Data Bits :7,8
	Flow Control : None, RTS/CTS, XON/XOFF
	Speed : up to 230Kbps
Input Voltage	DC 5 Volt
Power Consumption	Under 180 mA
Humidity	10 persen – 90 persen



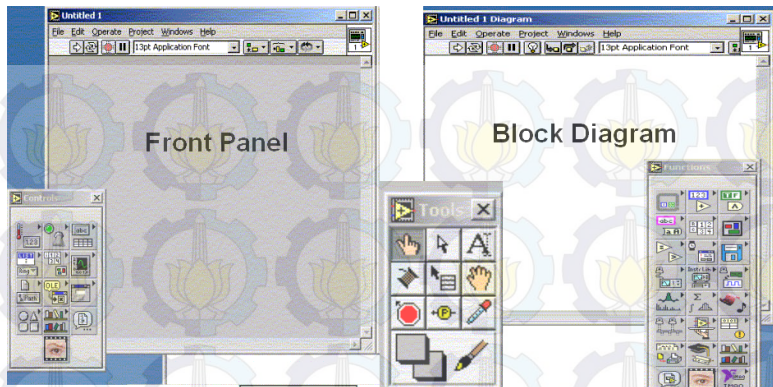
Gambar 2.7 Ethernet Tipe W1Z110SR

2.7 LabVIEW

LabVIEW adalah berorientasi objek grafis bahasa komputer yang dikembangkan untuk memfasilitasi perangkat keras dan perangkat lunak. Komunikasi *LabVIEW* adalah bahasa komputer yang lengkap yang dapat digunakan seperti *Basic*, *FORTRAN*, atau *C*. Dalam beberapa tingkatan instrumentasi *virtual* akuisisi data yang membuatnya mudah untuk mengontrol aliran data, dan pemrosesan sinyal banyak dan algoritma analisis datang dengan perangkat lunak sebagai instrumentasi *virtual premade*. Di dalam kelas, kesamaan antara instrumen *virtual* dan nyata membantu mahasiswa memahami bagaimana informasi dilewatkan antara komputer dan instrumen terpasang.

Program pada *LabVIEW* yang disebut sebagai VI (*Virtual Instrument*). Setiap VI menggunakan fungsi-fungsi yang menggerakkan masukan dari pemakai antarmuka atau sumber lain dan menampilkan informasi itu atau memindahkannya ke file lain atau ke komputer lain. VI menyediakan tiga komponen yaitu :

Panel muka (*front panel*), yang melayani antarmuka pengguna. *Front panel* merupakan *interface* antar pengguna (*user*) dengan program. Di dalam *front panel* terdapat kontrol (*input*) dan indikator (*output*) yang dikenal dengan Kontrol Palet. Kontrol pada *front panel* dapat berupa knop, tombol, dial, dan lainnya. Sedangkan untuk indikator (*output*) dapat berupa LED, grafik, dan tampilan lainnya. Kontrol menirukan input instrumen dan menyuplai data ke diagram blok pada VI yang bersangkutan. Indikator menirukan instrumen keluaran dan menampilkan data. Untuk menampilkan Kontrol Panel maka pada *front panel* dapat diaktifkan View >> Control Palette



Gambar 2.8 Tampilan *Front panel* dan *Block Diagram* Pada LabVIEW

Diagram blok, berisi kode sumber grafis yang menggambarkan fungsi-fungsi VI. Setelah membangun *front panel*, perlu menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan objek - objek *front panel*. Blok diagram berisi *source code* grafis ini. Objek - objek *front panel* tampak seperti terminal pada diagram blok. *Virtual Instrument* menunjukkan beberapa objek diagram blok seperti terminal - terminal, fungsi - fungsi dan alur. Pada diagram blok terdapat *Function Palette*, yang menyediakan *visible items* untuk membentuk suatu program. Jika tidak ditampilkan pada diagram blok *windows* maka dapat diaktifkan melalui *Window >> Show Functions Palette*, atau klik kanan area pada diagram *window*. Pada Gambar 2.8 merupakan tampilan *front panel* dan diagram blok pada LabVIEW [10].

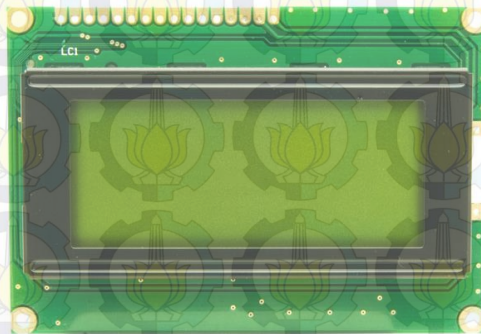
2.8 LCD 20 x 4

LCD (*Liquid Crystal Display*) adalah modul yang dapat menampilkan visual secara elektronik melalui cairan *liquid* yang terdapat pada modul tersebut.

LCD 20 x 4 adalah LCD yang terdiri dari 16 karakter dengan 4 baris sebagai *display* nya. Yang mana LCD ini memunculkan teks berwarna hitam dengan latar belakang (*background*) LCD adalah hijau. LCD 20 x 4 memiliki dimensi sebesar 87,0 x 60,0 x 13,6 mm [11].

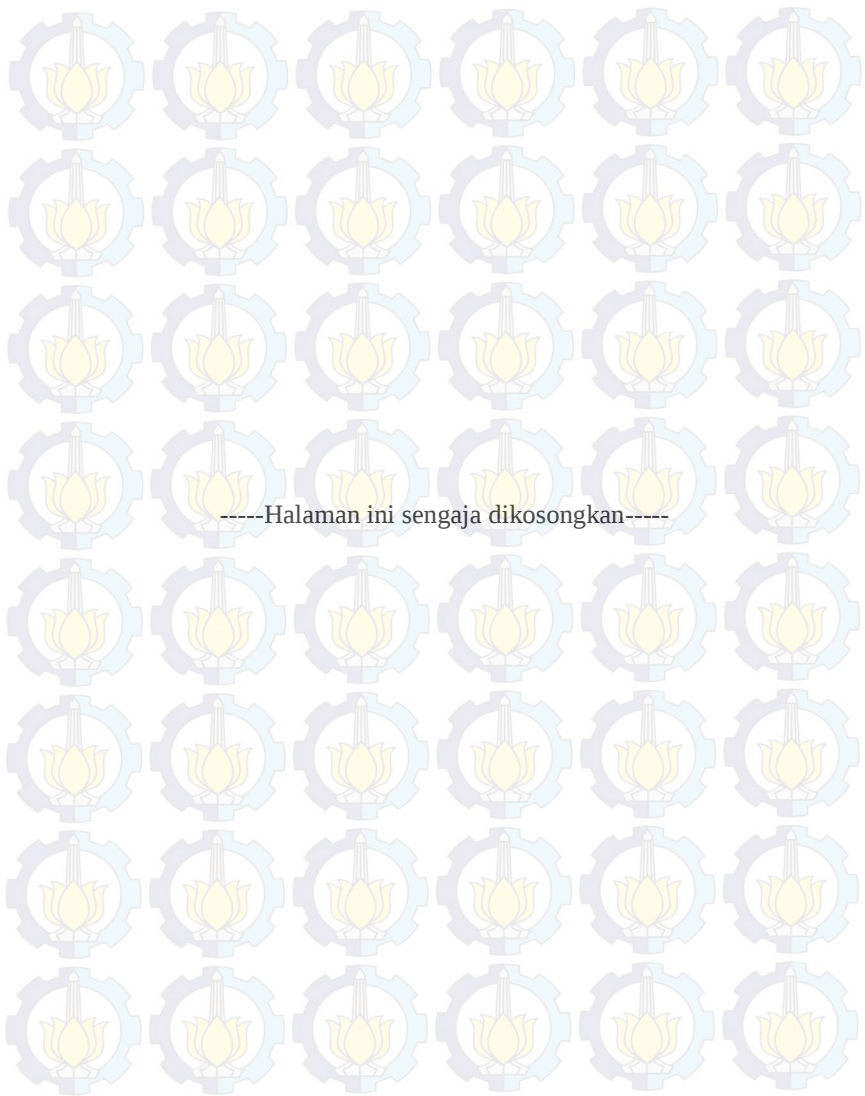
Tabel 2.3 Konfigurasi Pin LCD

No	Symbol	Function
1	Vss	<i>GND pin, 0V</i>
2	Vdd	<i>Positive power pin, +5</i>
3	Vo	<i>LCD drive voltage input pin</i>
4	Rs	<i>Data/Instrunction select input pin</i>
5	R/W	<i>Read/Write select input pin</i>
6	E	<i>Enable input pin</i>
7-14	D0 – D7	<i>Data bus line</i>
15	Led A	<i>LED power supply</i>
16	Led K	<i>LED power supply</i>



Gambar 2.9 LCD 20 x 4

Pada Gambar 2.9 merupakan modul LCD 20 x 4 karakter yang akan digunakan dalam Tugas Akhir ini. Salah satu alasan mengapa modul LCD dipakai dalam proyek akhir ini adalah untuk menunjukkan angka pengukuran daya kamar kost. Dengan mikrokontroler, dapat mengendalikan suatu peralatan agar dapat bekerja secara otomatis. Untuk mengakses LCD 20x4 harus melakukan konfigurasi pin dari LCD dengan pin I/O mikrokontroler tersebut. Pada Tabel 2.3 adalah deskripsi pin LCD.



BAB III

PERANCANGAN SISTEM KONTROL

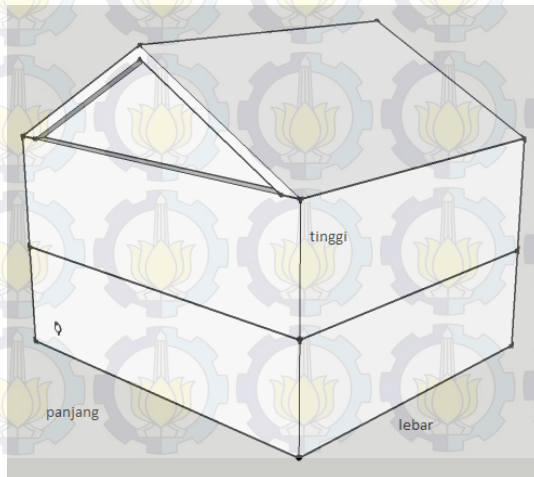
Pada bab ini akan dibahas mengenai perancangan alat yang terdiri dari perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

Perancangan perangkat keras (*hardware*) dibagi menjadi 2 bagian, yaitu meliputi perancangan elektronik dan juga perancangan mekanik. Perancangan elektronik terdiri dari perancangan sensor arus, sensor tegangan, sensor beda fasa yang menggunakan rangkaian *Zero Crossing Detector*, perancangan sistem komunikasi menggunakan *Ethernet* yang dilengkapi dengan rangkaian MAX232.

Untuk perancangan perangkat lunak (*software*) meliputi perancangan pemrograman mikrokontroler ATmega16, perancangan pemrograman modul *Ethernet*, perancangan *LabVIEW*.

3.1 Miniatur Rumah Kost

Miniatur rumah kost berfungsi sebagai tempat dari semua perangkat keras (*hardware*) diletakkan dan juga digunakan untuk menampilkan bagaimana proses kerja dari sistem yang dibuat. Gambar 3.1 merupakan sketsa dari miniatur rumah kost yang telah dirancang dan dibuat. Didalam miniatur ini terdapat dua buah kamar.



Keterangan :

P = 36 cm

L = 32 cm

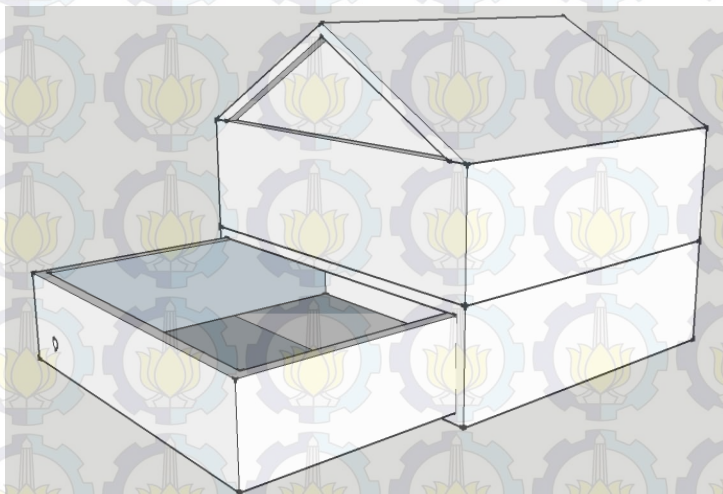
T = 20 cm

Gambar 3.1 Sketsa Miniatur Rumah Kost

Pada miniatur rumah kost ini didesain berupa bangunan dua lantai dengan panjang 36 cm, lebar 32 cm, tinggi 20 cm, dan pelengkap berupa genteng dengan tinggi kurang lebih 10 cm. Desain genteng yang berfungsi sebagai pelengkap tersebut, dapat dibuka agar lantai dua bisa terlihat seperti pada Gambar 3.3.

Pada lantai satu didesain lantai dasar bangunan dapat di buka untuk memasukkan *hardware* yang dibuat sebelumnya. Pada ujung lantai satu diberi satu lubang untuk menyambungkan kabel dari *hardware* ke stop kontak diluar, agar rangkaian bisa tersambung ke listik, diameter lubang sekitar 1 cm.

Pada lantai dua didesain ruangan berukuran sama yaitu dengan panjang 36 cm, lebar 32 cm, dan tinggi 10 cm. Tetapi pada lantai dua dibagi menjadi dua ruangan, untuk membedakan kamar satu dan kamar dua. Pada tiap kamar diberi 3 slot stop kontak, agar dapat tersambung beban yang ada, beban yang digunakan adalah beban lampu. Pada bagian ujung lantai dua, diberi dua buah lubang tiap kamar untuk menyambungkan kabel stop kontak ke *hardware* yang berada di lantai satu, diameter lubang kurang lebih 1 cm.

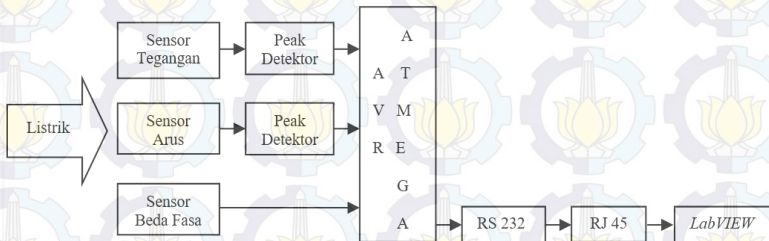


Gambar 3.2 Sketsa Lantai 1



Gambar 3.3 Sketsa Lantai 2

3.2 Blok Fungsional Sistem



Gambar 3.4 Blok Fungsional

Pada Gambar 3.4 merupakan blok fungsional sistem Rancang Bangun Alat Penghitung Penggunaan Daya dan Biaya Penggunaan Listrik Pada Kamar Kost Menggunakan *Ethernet* Sebagai Media Transmisi Data.

Setelah mengetahui gambaran umum sistem, maka dilanjutkan dengan perancangan sistem.

1. Perancangan dimulai dengan pembacaan tegangan jala-jala. Karena kemampuan mikrokontroler yang hanya bisa membaca tegangan DC maksimum 5 volt, maka dibutuhkan *step down transformer* dan rangkaian *peak detector*.

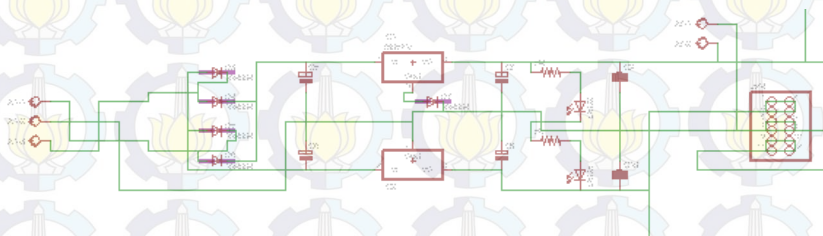
2. Perancangan berikutnya adalah membaca arus beban. Karena mikrokontroler hanya bisa membaca tegangan dan maksimum 5 volt, maka dibutuhkan trafo arus (*current tranformator*) dan rangkaian *peak detector*.
3. Setelah sensor tegangan dan sensor arus dirancang, berikutnya adalah merancang sensor beda fasa, sensor yang digunakan adalah *Zero Crossing Detector*.
4. Perancangan data pemakaian daya bisa di-monitoring dari LabVIEW menggunakan RS 232 yang disambung dengan kabel RJ45 dan dapat diakses melalui PC.

3.3 Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras pada Tugas Akhir ini terdiri dari perancangan *power supply*, sistem minimum ATmega16, perancangan tiga rangkaian sensor, yaitu sensor arus, sensor tegangan, sensor beda fasa yang menggunakan rangkaian *Zero Crossing Detector*.

Perancangan tiga rangkaian sensor tersebut dibutuhkan karena ATmega16 tidak mampu memproses tegangan dan arus yang berasal langsung dari PLN. ATmega16 hanya mampu diberi *input* tegangan sebesar 0VDC hingga 5VDC saja. Daya yang digunakan PLN untuk diakumulasikan dengan biaya per satu KWh-nya adalah daya aktif. Pada daya aktif terdapat unsur *Cos Phi* sebagai pengali. *Cos Phi* tersebut diperoleh dari perbedaan sudut antara tegangan dengan arus. Untuk mendapatkan nilai *Cos Phi*, digunakan fitur *timer counter* ATmega16 untuk mengetahui besar sudut fasanya

3.3.1 Perancangan Power Supply



Gambar 3.5 Rangkaian Power Supply

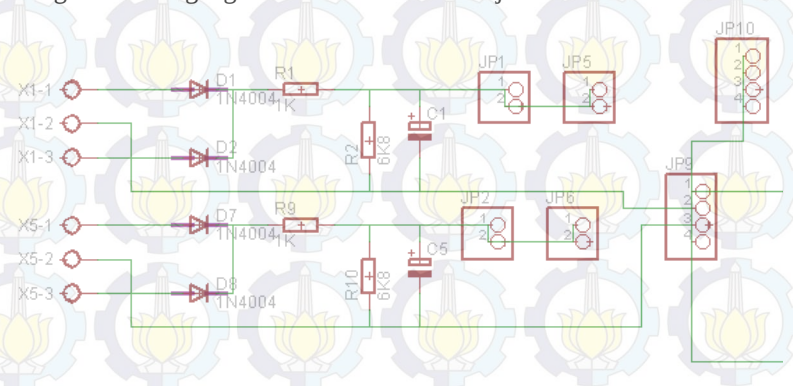
Power Supply ini digunakan untuk merubah tegangan VAC dari PLN sebesar 220 V menjadi tegangan searah (VDC) yang aman untuk rangkaian mikrokontroler serta rangkaian elektronik lainnya yang juga membutuhkan Tegangan *input* VDC.

Tegangan Jala-Jala PLN sebesar 220 V diturunkan oleh Trafo atau dari 220 VAC menjadi 12VAC. Karena Tegangan *output* dari Trafo masih berbentuk Tegangan VAC, maka dibutuhkan dua buah dioda untuk merubah menjadi gelombang *halfwave*, kemudian masuk ke rangkaian *power supply* yang terdiri dari kapasitor dan regulator. Perancangan rangkaian *Power Supply* dapat dilihat pada Gambar 3.5.

Power Supply digunakan untuk memberikan masukan tegangan pada beberapa komponen yang terdapat pada sistem alat ini, yaitu pada rangkaian sistem minimum mikrokontroler ATmega16 yang dimana rangkaian tersebut membutuhkan +5Vdc dan ground. Sedangkan IC sensor ZCD membutuhkan tegangan +5 Vdc dan -5 Vdc. Oleh karena itu, *power supply* yang digunakan mempunyai *output* +5 Vdc -5Vdc dan ground.

3.3.2 Sensor Tegangan

Sensor tegangan merupakan alat yang digunakan untuk mendeteksi besar tegangan yang melalui suatu peralatan listrik. Karena Atmega tidak bisa mendeteksi besar tegangan yang masuk kerumah kost yakni sebesar +- 220VAC, maka dibutuhkan sensor tegangan yang dapat mengkonversi tegangan dari +-220 VAC menjadi 0 – 5 VDC.



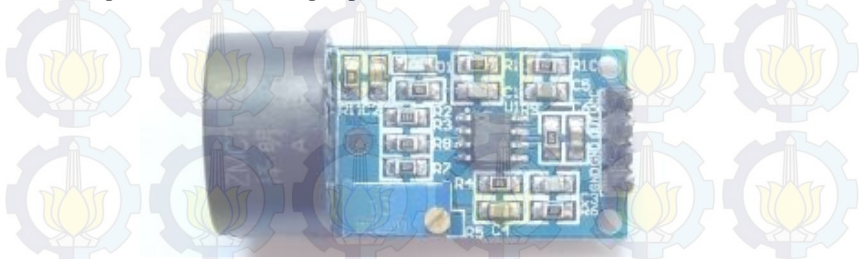
Gambar 3.6 Rangkaian Simulasi Sensor Tegangan

Sensor tegangan pada Tugas Akhir ini menggunakan *step-down transformer* 350mA untuk menurunkan nilai tegangan kerja yang digunakan. Tegangan *output* yang dihasilkan sebesar 6V. Cara kerja sensor tegangan pada Gambar 3.6 adalah keluaran tegangan *transformator step-down* 350mA masuk ke dioda agar *output* berubah menjadi *halfwave*, setelah itu *output* masuk ke rangkaian *peak detector* yang bertujuan untuk mendeteksi puncak *output* yang dihasilkan. Selanjutnya, *output* diberi kapasitor untuk mengurangi *noise* yang dihasilkan, agar *output* yang dihasilkan berupa DC. Keluaran dari kapasitor masuk ke *input* ADC mikrokontroler.

3.3.3 Sensor Arus

Sensor arus merupakan alat yang digunakan untuk mendeteksi besar arus yang melalui suatu peralatan listrik. Untuk mengubah besaran arus menjadi besaran tegangan yang bisa dibaca oleh mikrokontroler. Pada percobaan pertama, menggunakan sensor arus *simple phase AC* 5A. *Output* sensor tersebut sudah siap di *input* ke sistem ADC (*Analog to Digital Converter*) jadi tidak memerlukan rangkaian *peak detector*.

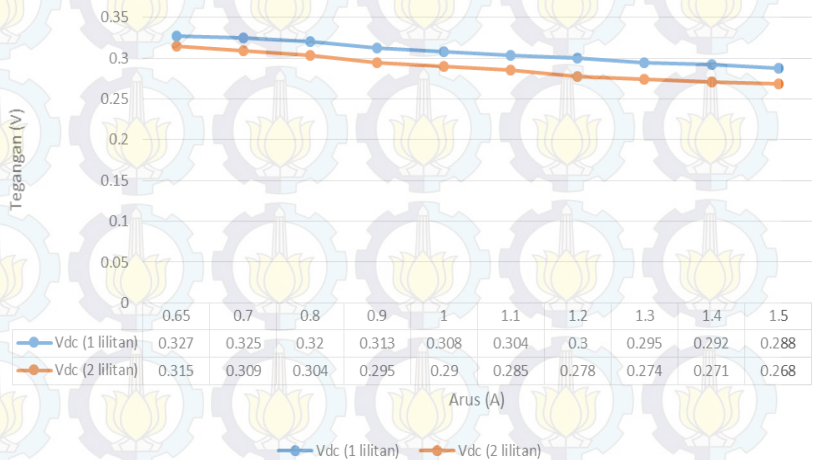
Pertama proses pengambilan data yang dihasilkan sensor arus *simple phase AC* 5A dengan cara memasukkan lilitan kabel dengan dimensi 0,75mm di toroidnya, tegangan referensi yang diberikan pada sensor diatas sebesar 3,3 Volt. Selanjutnya ADB Load dengan resistansi 320 Ω dan arus maksimal 1,5 A digunakan sebagai bebannya. Variasi lilitan kabel diperlukan untuk mengetahui fungsi perbedaan data oleh jumlah lilitan kabel yang dililitkan pada toroid. Variasi pertama menggunakan 1 lilitan dan variasi kedua menggunakan 2 lilitan kabel. Dibawah ini merupakan tabel data dan penampilan grafik yang diperoleh dari *output* sensor arus *simple phase AC* 5A.



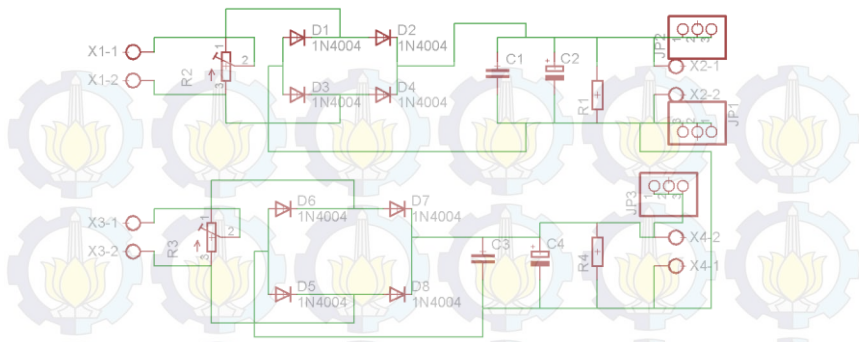
Tabel 3.1 Data Sensor Arus *Simple Phase AC 5A*

Arus	Vdc (1 lilitan)	Vdc (2 lilitan)
0,65	0,327	0,315
0,7	0,325	0,309
0,8	0,32	0,304
0,9	0,313	0,295
1	0,308	0,29
1,1	0,304	0,285
1,2	0,3	0,278
1,3	0,295	0,274
1,4	0,292	0,271
1,5	0,288	0,268

Grafik Sensor Arus Menggunakan ADB Load dengan Resistansi $320\ \Omega$ dengan arus maksimal 1,5 A



Gambar 3.8 Grafik Sensor Arus *Simple Phase AC 5A*



Gambar 3.9 Rangkaian Simulasi Sensor Arus

Data dari Sensor Arus *Simple Phase AC 5A* mengalami penurunan bila beban semakin besar, walaupun probabilitas kutub + dan – tidak tertukar. Data tersebut juga memberi kesimpulan bahwa variasi lilitan kabel berpengaruh pada keakuratan data yang diambil, bila lilitan kabel semakin banyak, maka data yang diperoleh semakin akurat dan linier.

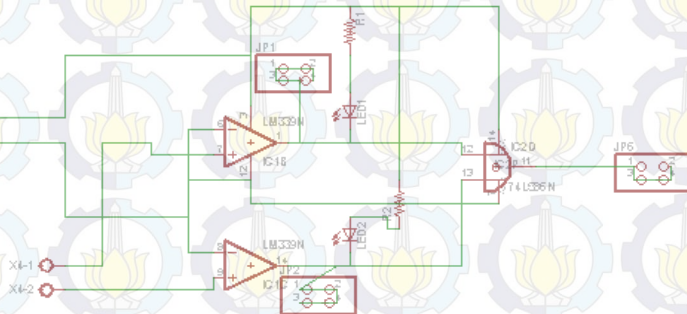
Dengan kesimpulan data yang diambil, sensor tersebut tidak digunakan sebagai sensor arus pada Tugas Akhir ini dikarenakan data yang dihasilkan tidak sesuai dengan kriteria yang digunakan, dan data sheet dari sensor tersebut juga sulit ditemukan, padahal data sheet adalah dasar dari cara penggunaan sensor tersebut.

Pada percobaan kedua, digunakan *current transformer* 5A/2,5mA seperti Gambar 2.3. Cara kerja rangkaian sensor arus pada Gambar 3.9 adalah *output current transformer* 5A/2,5mA yang diberi beban bervariasi dari 0 - 5A lalu masuk ke potensiometer 1k, agar *output* tegangan yang berupa AC bisa disesuaikan, selanjutnya *output* masuk ke dioda agar keluaran *output* menjadi *halfwave*. Setelah proses tersebut *output* masuk ke rangkaian *peak detector* yang bertujuan untuk mendeteksi puncak *output*. Keluaran *output* di beri kapasitor untuk mengurangi *noise* yang dihasilkan. Keluaran dari kapasitor masuk ke *input* ADC mikrokontroler.

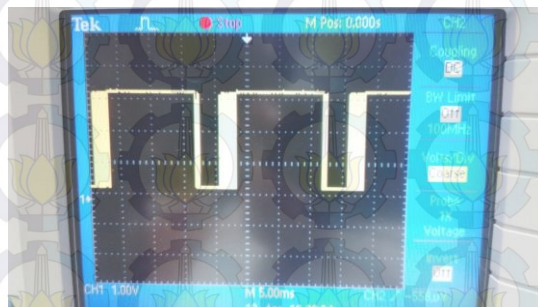
3.3.4 Perancangan Zero Crossing Detector

Sensor *Zero Crossing Detector* ini berfungsi untuk mendapatkan nilai *Cos Phi* yang diperoleh dari perbedaan sudut antara tegangan dengan arus. Pada percobaan pertama digunakan rangkaian seperti Gambar 3.10, cara kerjanya adalah *output* sensor tegangan dan sensor

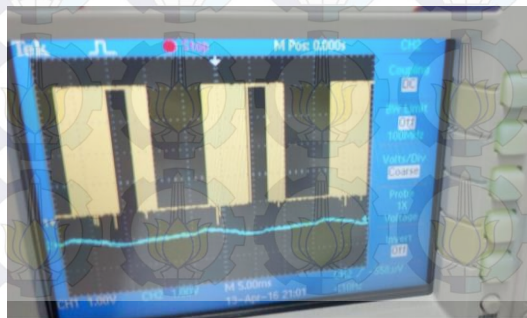
arus masuk ke komparator yakni LM339 yang digunakan untuk mengubah sinyal *output* menjadi sinyal kotak. Kemudian sinyal step dari sensor tegangan dan sensor arus dimasukkan gerbang logika XOR untuk menghasilkan sinyal *step* yang menunjukkan nilai beda *fasa*.



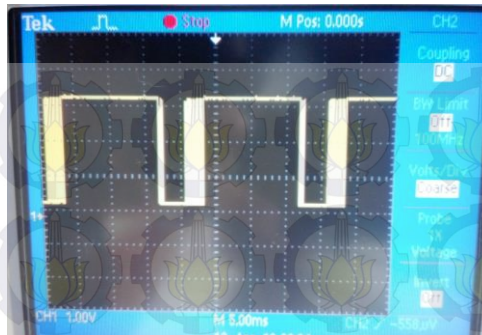
Gambar 3.10 Rangkaian Zero Crossing Detector Percobaan 1



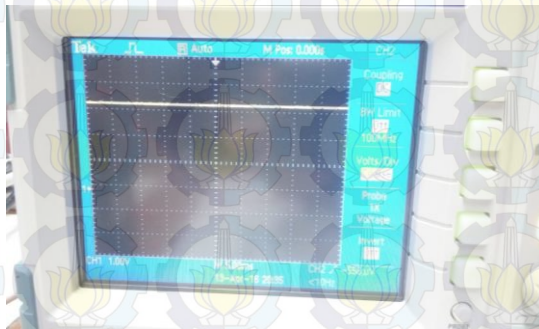
Gambar 3.11 Output Sensor Tegangan



Gambar 3.12 Output XOR



Gambar 3.13 Output Sensor Arus

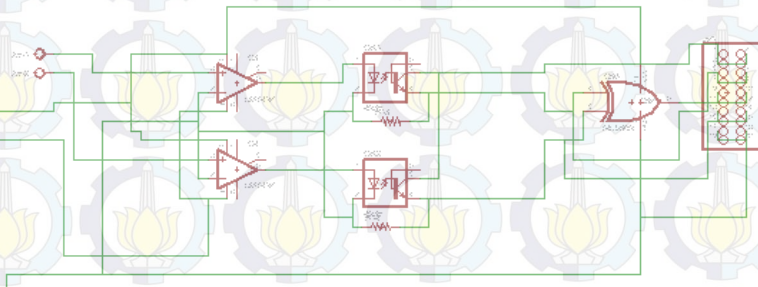


Gambar 3.14 Output XOR Saat Tersambung Ground

Percobaan pertama rangkaian *zero crossing detector* menggunakan LM 339 sebagai komparator yang mengolah *output* sensor tegangan dan sensor arus menjadi sinyal kotak. Pada Gambar 3.12 merupakan *output* dari XOR yang menghasilkan nilai beda fasa, tetapi pada Gambar 3.12 terlihat sinyal terdapat *noise*, yang menyebabkan sinyal terlihat kurang bagus. Kegagalan dari rangkaian pada Gambar 3.10 adalah saat *output* XOR dihubungkan oleh *ground* maka *output* XOR akan hilang seperti terlihat pada Gambar 3.14.

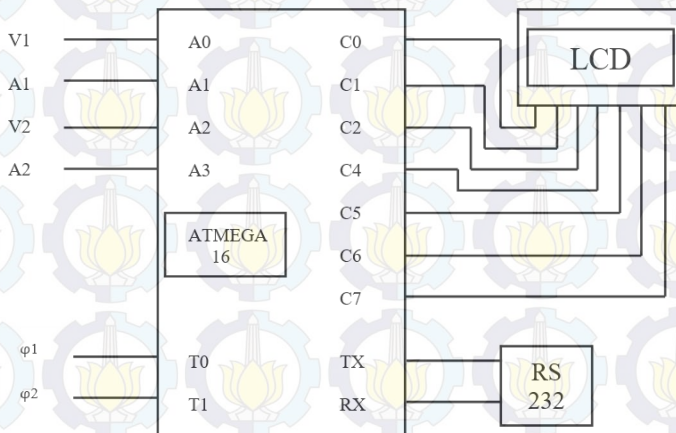
Karena ketidakcocokan data pada percobaan pertama, maka percobaan kedua dilakukan. Pada percobaan kedua, sensor *zero crossing detector* memanfaatkan *input* dari sensor tegangan dan sensor arus yakni di dapat dari keluaran transformator, untuk mengubah sinyal sinus menjadi sinyal kotak, maka digunakan komparator LM741.

Input dari komparator yang *inverting* diberi tegangan 0 volt atau *ground*. Sedangkan *input* non-invertingnya terhubung dengan sensor arus. Sehingga, komparator akan membandingkan tegangan *output* sensor arus dengan tegangan nol volt. *Output* dari komparator masih berupa tegangan 5 volt sampai (-5) volt. Oleh karena itu, diberi *optocoupler* agar sesuai dengan tegangan *input* ATmega16 yaitu 0-5 volt. Selain itu, *optocoupler* juga digunakan sebagai pelindung ATmega16 dari tegangan lebih. Kemudian *output* dari kedua *optocoupler* tadi dimasukkan ke *input* IC 74LS86 yang digunakan untuk mendapatkan perbedaan fasa.



Gambar 3.15 Rangkaian Zero Crossing Detector Percobaan 2

3.3.5 Perancangan Rangkaian ATmega16



Gambar 3.16 Perancangan Rangkaian Minimum ATmega16

Sebagai pengolah data, menggunakan mikrokontroler ATmega16. Mikrokontroler memiliki pin ADC sehingga dapat digunakan untuk mengkonversikan tegangan *analog* menjadi tegangan *digital*. Setelah tegangan diterima dan diolah dalam mikrokontroler maka hasil pengolahan tersebut akan ditampilkan kedalam LCD dan *LabVIEW*.

Penggunaan masing – masing port I/O mikrokontroler dalam sistem ini adalah sebagai berikut :

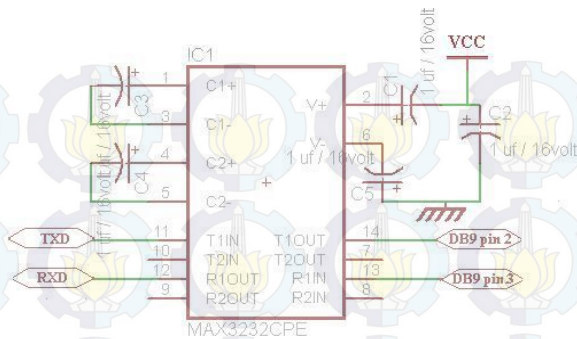
- a. *Port A.0 – Port A.3* sebagai masukan tegangan analog dari sensor.
- b. *Port C* sebagai keluaran ke LCD
- c. *Port TX dan Port RX* sebagai komunikasi dengan RS 232
- d. *Port T.0 – Port T.1* sebagai timer

Tegangan analog pada sensor akan masuk melalui mikrokontroler melalui *port A.0 – A.3*. Setelah dikonversi menjadi keluaran digital yang berupa biner, akan ditampilkan pada LCD berupa jumlah watt *hour* (daya dalam satuan detik) yang telah dengan mikrokontroler melalui *port C*. *Port TX dan RX* digunakan dalam sistem komunikasi data yaitu RS 232 yang dihubungkan oleh *Ethernet*. Dan yang terakhir, port T.0 – T.1 yang digunakan sebagai *inputan* dari *output* sensor beda fasa.

3.3.6 Perancangan Rangkaian MAX232

Pada Mikrokontroler ATmega16 terdapat pin untuk komunikasi TX dan RX. TX digunakan untuk mengirimkan data secara serial sedangkan RX digunakan untuk menerima data serial. Komunikasi data serial pada mikrokontroler ini masih menggunakan sinyal TTL atau *Transistor Transistor Logic* yaitu sinyal yang gelombang datanya antara 0 dan 5V. Dengan fasilitas RX dan TX ini mikrokontroler bisa komunikasi secara serial baik antara *device* atau dengan komputer.

Jika ingin digunakan untuk berkomunikasi antar *device* maka hubungkan *cross* antar dua *device*, yaitu TX *device* 1 dihubungkan ke RX *device* 2 dan RX *device* 1 dihubungkan ke TX *device* 2. Sedangkan jika ingin digunakan untuk berkomunikasi dengan komputer maka RX dan TX tidak bisa langsung dihubungkan begitu saja karena sinyal yang digunakan berbeda. Komunikasi serial komputer menggunakan sinyal RS232 yaitu sinyal yang gelombangnya antara +25 V sampai -25 V. Oleh karena itu jika ingin berkomunikasi antara Mikrokontroler dengan komputer dibutuhkan sebuah komponen yang dapat mengubah sinyal level TTL dari mikrokontroler menjadi sinyal level RS232. Salah satu komponen yang sering digunakan adalah IC MAX232.



Gambar 3.17 Rangkaian MAX 232

Supaya dapat digunakan dengan baik maka IC MAX232 ini membutuhkan beberapa komponen tambahan yaitu Kapasitor elektrolit (ELCO) sebanyak 4 buah sebesar 1uF. Untuk lebih lengkapnya rangkaian MAX232 dapat dilihat pada Gambar 3.17.

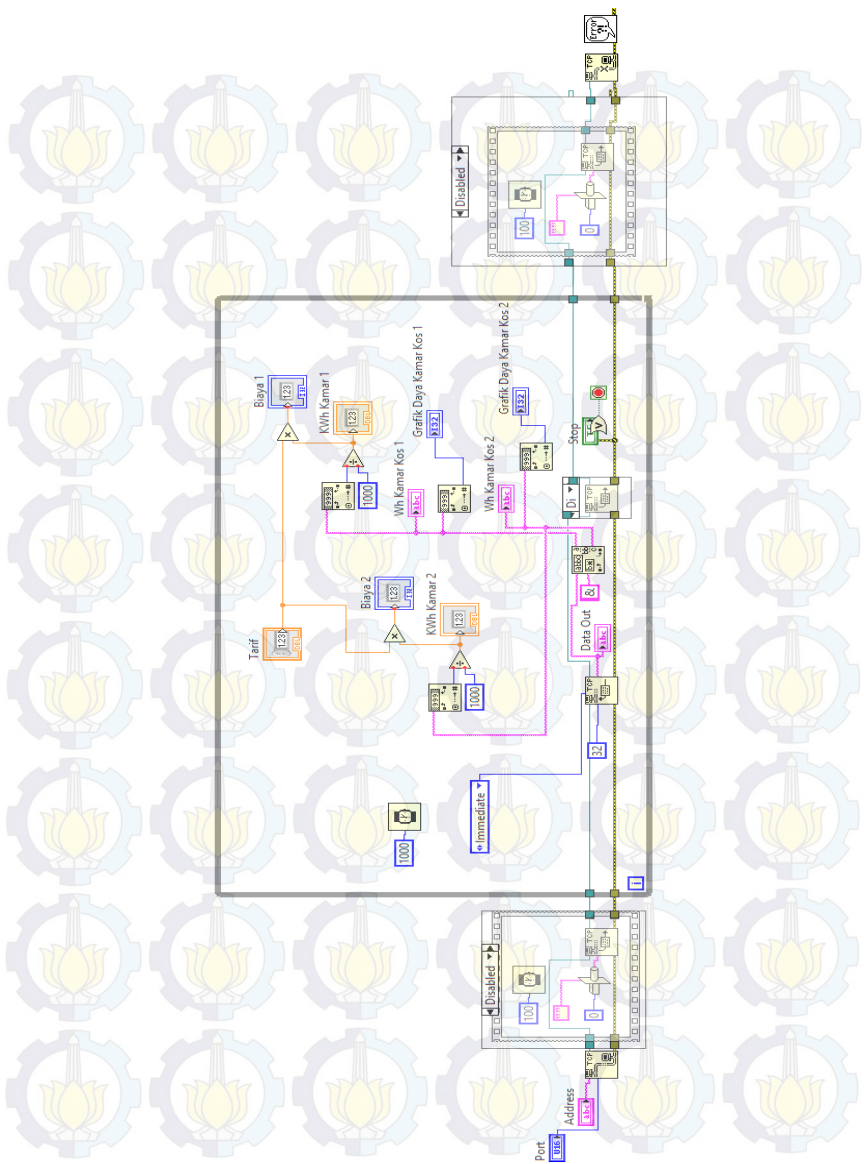
TXD dan RXD dihubungkan ke minimum sistem mikrokontroler ATmega16 sebagai *interface* ke komputer digunakan konektor serial DB9, konektor ini memiliki 9 buah pin, tetapi yang digunakan cukup 2 saja yaitu Td, Rd, dan Gnd. Konektor DB9 ini yang kemudian akan disambungkan ke *Ethernet WIZ110SR*.

3.4 Perancangan Perangkat Lunak (Software)

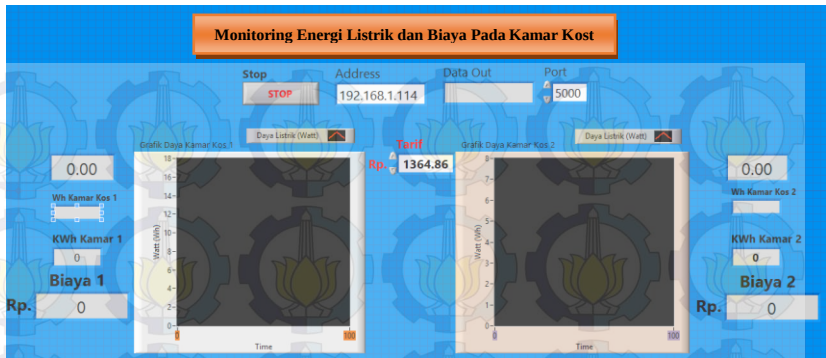
Perangkat lunak digunakan untuk media *LabVIEW* yang menampilkan hasil *monitoring* penggunaan daya tiap kamar, sedangkan untuk program yang dibuat pada mikrokontroler AVR ATmega16 dengan cara pemrograman menggunakan *software code vision AVR*.

3.4.1 Perancangan *LabVIEW*

Pada Tugas Akhir ini menggunakan *LabVIEW* sebagai *interface* untuk *monitoring* penggunaan daya pada kamar kost. Pada Gambar 3.19 adalah tampilan luar dari program *LabVIEW* didalamnya terdapat fitur *stop* untuk menghentikan program, *address* dan port untuk memasukkan alamat IP dan port Ethernet yang digunakan untuk media komunikasi data. tabel kWh merupakan *output* besar kWh yang digunakan, dan tabel biaya yang fungsinya untuk mengetahui biaya yang harus dibayarkan, dengan cara mengalikan besar kWh dan tarif yang *diinput* pada tabel tarif. Dan Pada Gamabr 3.18 merupakan blok diagram *LabVIEW*.



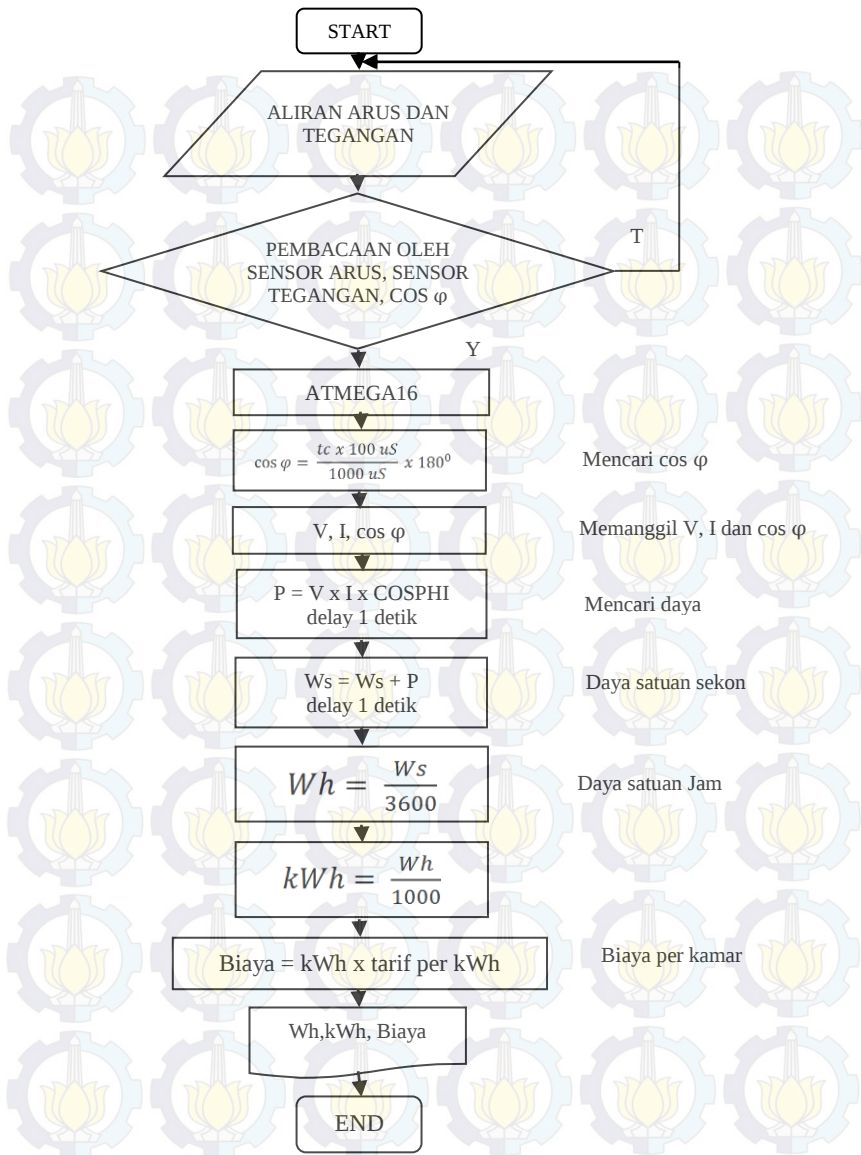
Gambar 3.18 Tampilan Blok Diagram Pada LabVIEW



Gambar 3.19 Tampilan *Front panel* Pada *LabVIEW*

3.4.2 Alur Program Pada Mikrokontroler

Berikut ini adalah alur program pada mikrokontroler ATmega16. Dimulai dengan mendeteksi adanya aliran arus dan tegangan, kemudian mengalami pembacaan oleh sensor arus dan sensor tegangan, bila pembacaan gagal, maka sistem akan mengulangi proses mendeteksi aliran arus dan tegangan, sedangkan bila pembacaan berhasil maka, sistem akan diproses oleh ATmega16 dengan memasukkan rumus $P = V \times I \times \cos \varphi$ untuk mendeteksi daya dan $Ws = Ws + P$ untuk sistem penjumlahan daya persatuan detik, setelah itu Daya per satuan detik dikonversi menjadi daya persatuan jam dengan membagi daya per satuan detik dibagi dengan 3600, karena 1 jam = 3600 sekon. Lalu daya per satuan jam dikonversi ke *kilo watt hours* dengan cara dibagi 1000. Untuk biaya, nilai kWh dikali dengan tariff per kWh, dengan menggunakan tariff PLN terbaru pada bulan Juni 2016 dengan harga 1354,68. Berikutnya, sistem akan menampilkan Wh, kWh dan biaya.



Gambar 3.20 Flow Chart Sistem

3.4.3 Perancangan *Ethernet*

Dalam penggunaan modul *Ethernet*, diperlukan suatu perangkat lunak (*software*) untuk dapat mengkonfigurasi melalui media komputer. Berikut penjelasan dari *software* WIZ110SR Configuration Tool yang terlihat pada Gambar 3.16.

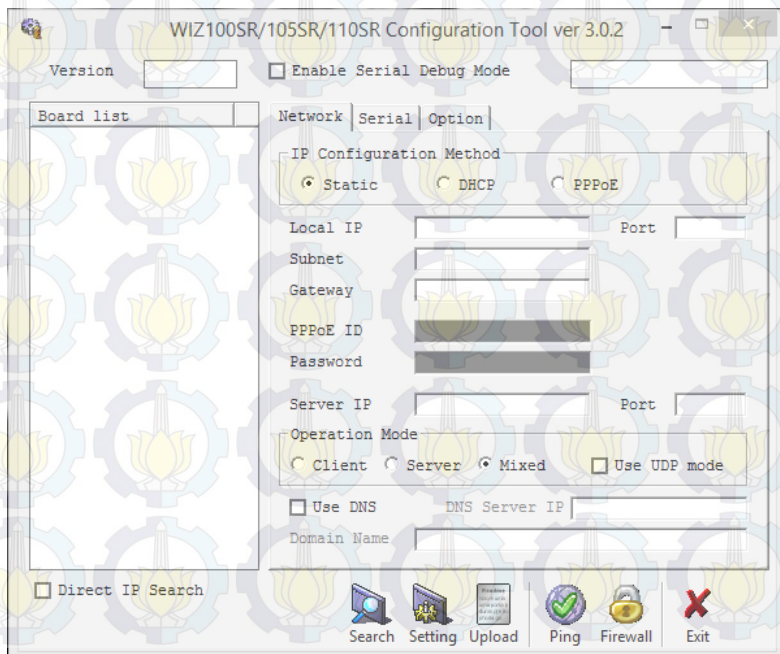
Keterangan mengenai fungsi atau kegunaan dari berbagai elemen pada *software*:

a. *Version*

Menampilkan versi *firmware* yang dipakai.

b. *Enable Serial Debug Mode*

Jika mode ini dipakai, dapat digunakan untuk memonitor status dan pesan WIZ110SR (mendengarkan OK, sambungkan gagal dll) melalui *port* serial. Jika modus ini diaktifkan, pesan debug dapat menyebabkan operasi yang tidak normal dari perangkat serial. Oleh karena itu, modus ini digunakan hanya untuk modus *Debug*.



Gambar 3.21 Tampilan *Software* WIZ110SR Configuration Tool

c. *Board List*

Jika tombol *search* di klik, semua alamat MAC pada *subnet* yang ada akan ditampilkan pada papan ini.

d. *Local IP/Port*

Alamat IP dari W1Z110SR dan nomor *port* untuk koneksi jaringan.

e. *Subnet*

Subnet mask dari W1Z110SR.

f. *Gateway*

Gateway address dari W1Z110SR.

g. *Server IP/Port*

Ketika W1Z110SR diatur sebagai mode *Client* atau mode *Mixed*, *server IP* dan *port* harus ditentukan. W1Z110SR mencoba untuk menyambungkan alamat IP ini.

h. *Use DNS*

Jika fungsi *DNS* diperlukan, cek pilihan ini dan masukan nama domain dari *DNS (Domain Name System)* adalah sistem *database* yang memiliki informasi tentang alamat IP dan nama domain yang sesuai. Jika ingin melakukan koneksi ke nama domain, dapat menggunakan fungsi *DNS* dari W1Z110SR. Masukan alamat IP dari *server DNS* yang disediakan oleh ISP dalam *DNS Server IP*, dan domain nama di *server (domain)*.

i. *Enable DHCP Mode*

Mode ini dipilih untuk menggunakan mode *DHCP*. Caranya pilih *Enable DHCP mode* dan klik *Setting button*. Jika IP address berhasil diperoleh dari *server DHCP*, alamat MAC yang akan ditampilkan pada *board list*. Ketika sebuah alamat di dalam daftar dipilih maka alamat IP, *Subnet mask* dan *Gateway* yang ditampilkan. Jika modul tidak bisa mendapatkan informasi jaringan dari *server DHCP*, alamat IP, *gateway address* dan *subnet mask* akan diinisialisasi ke 0.0.0.0.

j. *Network mode: client/server/mixed*

Digunakan untuk memilih metode komunikasi berbasis pada TCP – TCP *server*, TCP *Client*, dan *Mixed*. TCP merupakan protocol untuk membuat koneksi sebelum komunikasi data, tetapi UDP hanya digunakan untuk komunikasi data tanpa pembentukan jaringan.

k. *Use UDP Mode*

Pada mode *UDP*, pembentukan koneksi tidak didefinisikan. Hanya menetapkan alamat IP dan nomor *port* dari rekan dan mengirim data.

l. *Direct IP Search*

Digunakan untuk mencari W1Z110SR yang tidak di-*install* di subnet yang sama. Jika pencarian IP langsung diperiksa, alat konfigurasi menggunakan TCP UDP bukan disiarkan untuk mencari modul. Oleh karena itu, jaringan informasi dari alamat IP seperti modul, *subnet mask*, dan *gateway* diperlukan. Untuk mencari IP, masukan alamat IP yang akan diaktifkan. Dalam bidang ini, masukan alamat IP modul, dan klik tombol “Cari”.

m. *Search*

Fungsi pencarian digunakan untuk mencari semua modul yang ada pada LAN yang sama. Dengan menggunakan UDP siaran, semua modul di *subnet* yang sama akan dicari.

n. *Setting*

Fungsi ini adalah untuk menyelesaikan perubahan konfigurasi. Jika anda pilih alamat MAC, konfigurasi *default* nilai modul akan ditampilkan. Mengubah konfigurasi dan klik *setting* untuk menyelesaikan tombol konfigurasi. Modul akan menginisialisasi ulang dengan konfigurasi yang telah dirubah. Nilai konfigurasi dapat diubah dalam langkah – langkah dibawah ini :

1. Pilih alamat MAC yang menyatakan nilai konfigurasi
2. Ubah nilai
3. Klik tombol *setting*
4. Akan diinisialisasi dengan nilai - nilai yang dirubah

o. *Upload*

Firmware akan di-*upload* melalui jaringan.

p. *Exit*

Menutup program konfigurasi W1Z110SR.

Tata Cara Penggunaan Modul *Ethernet*

Dalam menggunakan modul *Ethernet* terdapat beberapa langkah yang harus dilakukan, diantaranya:

- a. Mempersiapkan modul *Ethernet*, rangkaian RS 232, kabel LAN dan kabel penghubung RS 232 ke mikrokontroler.

- b. Setelah semua alat terhubung, selanjutnya mengaktifkan *firmware* untuk modul yang berupa *software W1Z110SR configuration tool* pada komputer. Untuk men-*setting* penggunaan modul
- c. Setelah memberikan alamat pada modul, selanjutnya mengatur alamat IP pada komputer. Alamat ini harus sama dengan alamat modul.
- d. Setelah proses *setting* modul dan komputer selesai, selanjutnya dilakukan tes koneksi. Jika semua *setting* sudah benar, maka modul dapat digunakan.

BAB IV

HASIL SIMULASI DAN IMPLEMENTASI

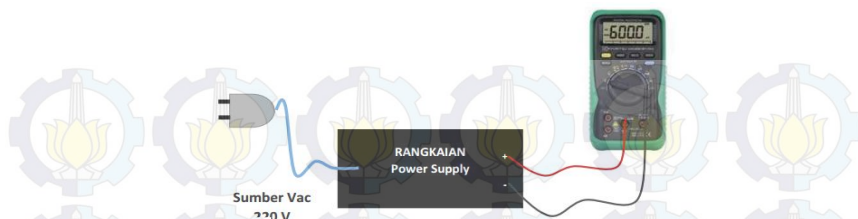
Pada bab ini akan membahas mengenai hasil pengujian dan analisa dari penyusunan alat untuk Alat Penghitung Penggunaan Daya Dan Biaya Penggunaan Listrik Pada Kamar Kost Menggunakan *Ethernet* Sebagai Media Transmisi Data. Pengujian alat ini ditujukan untuk memastikan agar peralatan yang dibuat dapat berfungsi dengan baik.

Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai pada pengujian sistem. Pengujian juga bertujuan untuk mengetahui kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan atau kesalahan dalam sistem. Pengukuran dan pengujian pada Tugas Akhir ini meliputi :

1. Pengukuran Tegangan *Output* Rangkaian *Power Supply*.
2. Pengujian Pin Mikrokontroler ATmega16.
3. Pengujian Rangkaian Sensor Tegangan.
4. Pengujian Rangkaian Sensor Arus.
5. Pengujian Rangkaian Sensor *Zero Crossing Detector*.
6. Pengujian Rangkaian Komunikasi.
7. Pengujian Beban Variasi.
8. Pengujian Keseluruhan.

4.1 Pengukuran Rangkaian *Power Supply*

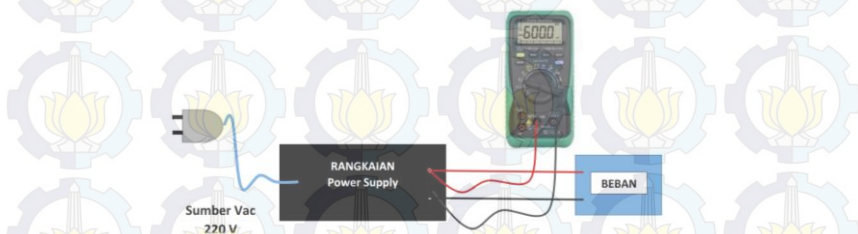
Pada Tugas Akhir ini perlu dilakukan pengujian rangkaian *power supply* guna mengetahui tegangan keluaran dari *power supply* apakah telah sesuai dengan tegangan *output* yang dikehendaki ataukah belum. Untuk cara pengujian menggunakan dua metode yaitu, pengujian *power supply* menggunakan beban dan pengujian *power supply* tidak menggunakan beban. Untuk pengukuran pertama, dilakukan pengukuran *power supply* tanpa dihubungkan ke beban. Konfigurasi pengukuran Rangkaian *Power Supply* tanpa beban dapat dilihat pada Gambar 4.1, Tabel 4.1 merupakan hasil dari pengujian *power supply* 5 V tanpa dihubungkan ke beban, dan Tabel 4.2 merupakan hasil dari pengujian *power supply* 5 V, -5 V dengan dihubungkan ke beban.



Gambar 4.1 Pengukuran Tanpa Beban

Tabel 4.1 Hasil Pengukuran *Power Supply* 5 Volt, -5 Volt Tanpa Beban

<i>Input Power Supply</i>	<i>Output Positif (Volt)</i>	<i>Output Negatif (Volt)</i>
225 V	5,68	-5,04
220 V	5,66	-5,04
210 V	5,65	-5,04
200 V	5,65	-5,04



Gambar 4.2 Pengukuran dengan Menggunakan Beban

Tabel 4.2 Hasil Pengukuran *Power Supply* 5 Volt dengan Beban

<i>Input Power Supply</i>	<i>Output Positif (Volt)</i>	<i>Output Negatif (Volt)</i>
225 V	5,62	-5,04
220 V	5,62	-5,04
210 V	5,62	-5,04
200 V	5,62	-5,04

Pengukuran selanjutnya yaitu dengan menghubungkan Rangkaian *Power Supply* ke beban berupa rangkaian mikrokontroler dan Sensor *Zero Crossing Detector* untuk *power supply* 5 V, -5 V. kemudian diukur kembali besar tegangan *output*nya apakah mengalami *drop* tegangan atau tidak. Konfigurasi pengukuran Rangkaian *Power Supply* dengan menggunakan beban dapat dilihat pada Gambar 4.2.

Dari data yang diperoleh diatas, *power supply* yang digunakan dapat berfungsi sebagai pensuplai daya yang dibutuhkan oleh keseluruhan sistem alat karena *output* dari *power supply* dalam *range* yang dibutuhkan.

4.2 Pengujian Pin Mikrokontroler

Mikrokontroler yang digunakan dalam alat ini yaitu menggunakan ATmega16. Untuk mengetahui apakah pin-pin pada mikrokontroler dapat digunakan dan tegangan yang dikeluarkan sesuai dengan *datasheet* yang ada. Pengukuran dilakukan dengan cara mengukur tegangan pada setiap pin mikrokontroler ATmega16 yang telah dipasang pada *minimum system* yang digunakan untuk mengecek apakah terjadi kesalahan pada *minimum system* yang telah dibuat. Dengan mula mula program di *upload* dari laptop ke mikrokontroler ATmega16 terlebih dahulu.

Program uji coba mikrokontroler dalam keadaan *High Voltage* (1):

```
.....
void setup() {
    // put your setup code here, to run once:
    digitalWrite(2,HIGH);
    digitalWrite(3,HIGH);
    digitalWrite(4,HIGH);
    digitalWrite(5,HIGH);
    digitalWrite(6,HIGH);
    digitalWrite(7,HIGH);
}
.....
```

Program ujicobamikrokontrolerdalamkeadaanLow Voltage (0):

```
.....
void setup() {
    // put your setup code here, to run once:
    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
}
.....
```

Gambar 4.3 Program Uji Coba Mikrokontroler

Tabel 4.3 Hasil Pengukuran Tegangan pada Setiap Pin Mikrokontroler

Pin	High (Volt)	Low (Volt)	Pin	High (Volt)	Low (Volt)
B0	4,98	0	D0	4,98	0
B1	4,98	0	D1	4,98	0
B2	4,98	0	D2	4,98	0
B3	4,98	0	D3	4,98	0
B4	4,98	0	D4	4,98	0
B5	4,98	0	D5	4,98	0
B6	4,98	0	D6	4,98	0
B7	4,98	0	D7	4,98	0
C0	4,98	0	-	-	-
C1	4,98	0	-	-	-
C2	4,98	0	-	-	-
C3	4,98	0	-	-	-
C4	4,98	0	-	-	-
C5	4,98	0	-	-	-
C6	4,98	0	-	-	-
C7	4,98	0	-	-	-
Rata – Rata	4,98	0	Rata – Rata	4,98	0

Setelah program di *upload* maka pengukuran dilakukan pada saat tegangan dalam posisi *High Voltage* dan *Low Voltage*. Tegangan *input* yang digunakan bersumber dari *USB* laptop yaitu sebesar +5 V. Hasil pengukuran tersebut dapat dilihat pada Tabel 4.3.

Dilihat dari data hasil pengukuran tersebut, mikrokontroler (ATmega16) apabila mendapat logika 1 maka tegangan *output* sebesar 4,89 V dan saat mendapat logika 0 maka tegangan *output* sebesar 0 Volt. Ini berarti Mikrokontroler ATmega16 yang akan digunakan dalam kondisi bagus dan dapat digunakan sesuai kebutuhan.

4.3 Pengujian Rangkaian Sensor Tegangan

Sensor tegangan merupakan alat yang digunakan untuk mengukur besar tegangan yang ada. Disini digunakan variasi tegangan menggunakan *auto transformer* untuk mengubah – ubah nilai tegangan.

Hasil pengukuran ditunjukkan pada Tabel 4.4 untuk kamar 1, variasi *input* tegangan dimulai dari 0 Volt – 240 Volt, dan grafik pada Gambar 4.4 yang memperjelas perubahan *output* sensor tegangan.

Pengambilan data untuk kamar 2 terlihat pada Tabel 4.5, variasi *input* tegangan dimulai dari 0 Volt – 240,7 Volt, grafik pada Gambar 4.5 yang memperjelas perubahan *output* sensor tegangan.

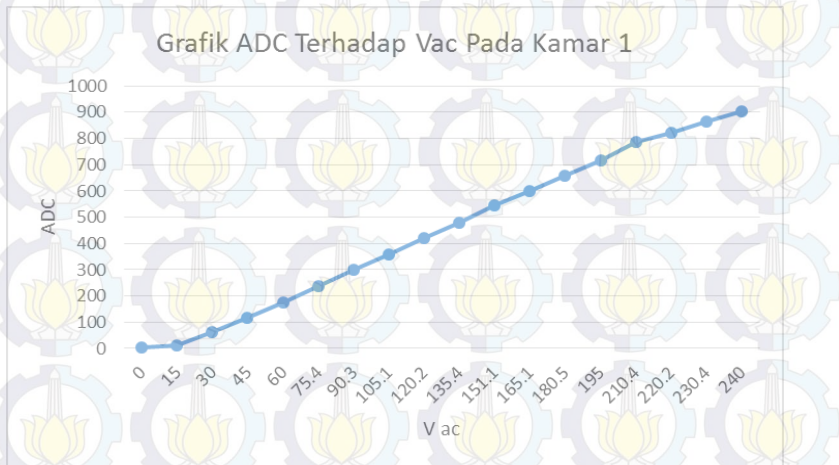
Tabel 4.4 Pengambilan Data Sensor Tegangan Kamar 1

Vac	ADC
0	4
15	13
30	63
45	116
60	175
75,4	237
90,3	300
105,1	359
120,2	419
135,4	479
151,1	544
165,1	600
180,5	660
195	718
210,4	785
220,2	823
230,4	865
240	902

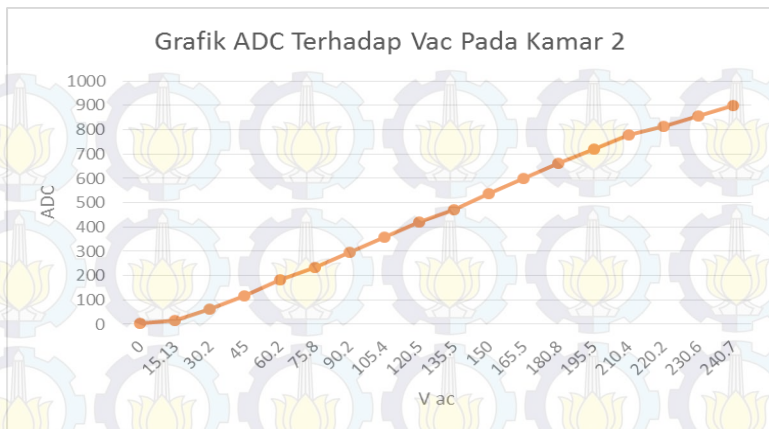
Tabel 4.5 Pengambilan Data Sensor Tegangan Kamar 2

Vac	ADC
0	4
15,13	14
30,2	62
45	118
60,2	182
75,8	234

Vac	ADC
90,2	295
105,4	358
120,5	422
135,5	473
150	539
165,5	600
180,8	663
195,5	720
210,4	778
220,2	816
230,6	856
240,7	898



Gambar 4.4 Grafik ADC Terhadap Tegangan Kamar 1



Gambar 4.5 Grafik ADC terhadap Tegangan Pada Kamar 2

Berdasarkan hasil pengujian kedua sensor tegangan diatas, *ouput* yang dihasilkan kedua sensor menunjukkan dalam *range* 0-5 VDC, oleh sebab itu, kedua rangkaian sensor tegangan tersebut dapat digunakan karena ADC dari mikrokontroler dapat memproses *output* dari kedua sensor. Namun masih diperlukan adanya linierisasi data agar data yang diproses di mikrokontroler mempunyai linieritas yang bagus. Karenanya, kedua sensor tegangan tersebut dapat digunakan sebagai implementasi tegangan yang diperoleh dari PLN.

4.4 Pengujian Rangkaian Sensor Arus

Rangkaian sensor arus ini menggunakan *current transformator* 5A/2,5mA. Hasil pengukuran ditunjukkan pada Tabel 4.6 dan Tabel 4.7, variasi *input* arus dimulai dari 0,37A – 4,9A, sebagai penjelas terdapat grafik pada Gambar 4.3 dan Gambar 4.4 yang terlihat perubahan *output* sensor arus. Pengambilan data melalui osiloskop dilakukan untuk mengetahui lebih jelas bentuk gelombang yang ada dan mengetahui ada atau tidaknya *ripple*.

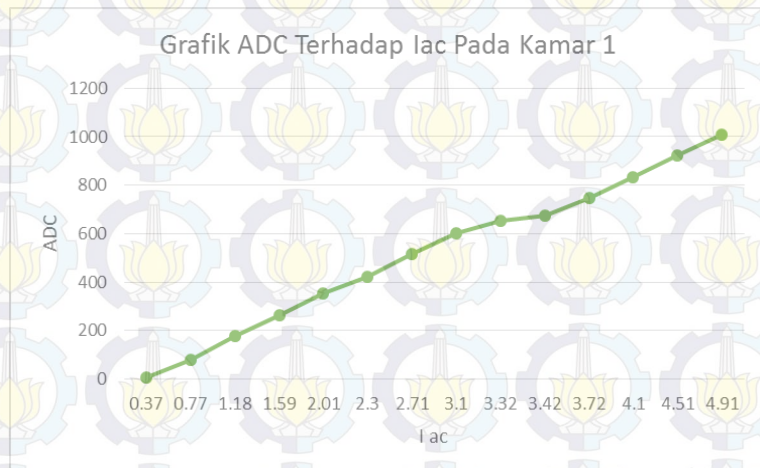
Tabel 4.6 Pengambilan Data Sensor Arus Kamar 1

Arus Input (A)	ADC
0,37	7
0,77	80
1,18	176
1,59	263
2,01	352
2,3	422
2,71	515
3,1	601
3,32	654
3,42	673
3,72	749
4,1	832
4,51	921
4,91	1009

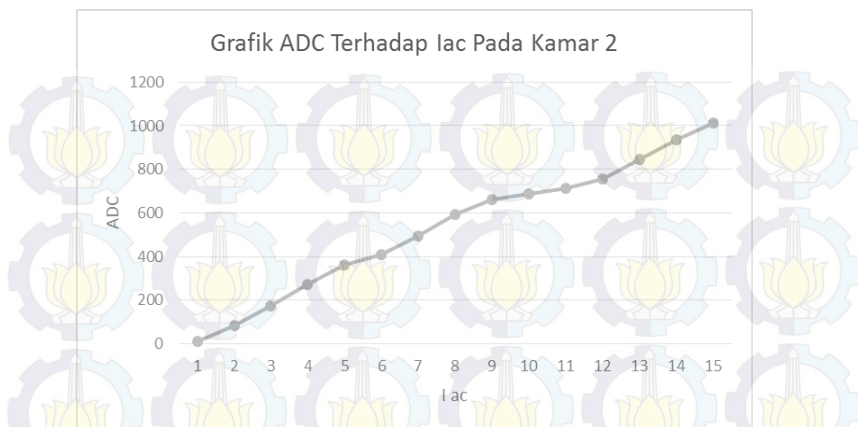
Tabel 4.7 Pengambilan Data Sensor Arus Kamar 2

Arus Input (A)	ADC
0,36	10
0,78	84
1,2	173
1,62	270
2,04	361
2,24	410

Arus Input (A)	ADC
2,65	496
3,07	595
3,36	660
3,48	688
3,57	713
3,77	754
4,16	847
4,57	936
4,97	1011



Gambar 4.6 Grafik Arus Terhadap ADC Pada Kamar 1



Gambar 4.7 Grafik Arus Terhadap ADC Pada Kamar 2

Berdasarkan hasil pengujian kedua sensor arus diatas, *ouput* yang dihasilkan kedua sensor menunjukkan dalam *range* 0-5 VDC, oleh sebab itu, kedua rangkaian sensor arus tersebut dapat digunakan karena ADC dari mikrokontroler dapat memproses *output* dari kedua sensor arus. Namun masih diperlukan adanya linierisasi data agar data yang diproses di mikrokontroler mempunyai linieritas yang bagus. Karenanya, kedua sensor arus tersebut dapat digunakan sebagai implementasi arus yang diperoleh dari beban masing-masing sensor.

4.5 Pengujian Rangkaian Sensor Zero Crossing Detector

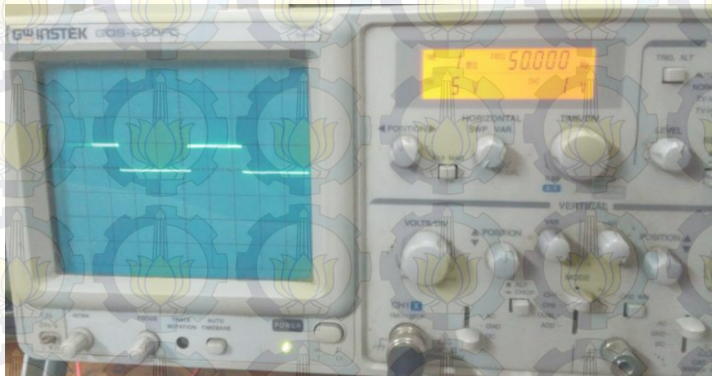
Rangkaian ini menggunakan ic XOR untuk mendapatkan perbedaan fasa. Ic XOR digunakan untuk mencari selisih beda fasa, seperti yang diketahui bahwa ic XOR memiliki cara kerja sebagai berikut:

1. Jika kedua *input* berlogika sama maka *output* akan bernilai logika "0".
2. Jika kedua *input* berlogika tidak sama maka *output* akan bernilai logika "1".

Dibawah ini terdapat beberapa gambar sinyal osiloskop yang diambil dari *output* sensor tegangan dan sensor arus pada kamar 1 dan 2. Selain itu, juga terdapat gambar sinyal osiloskop yang diambil dari *output* XOR pada kamar 1 dan 2.



Gambar 4.8 Output LM 741 dari Sensor Tegangan Pada Kamar 1

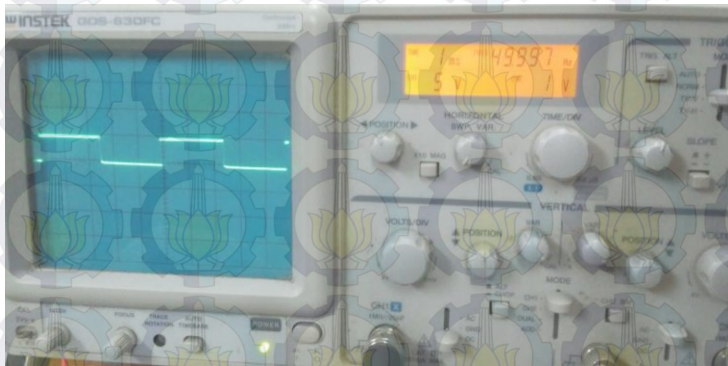


Gambar 4.9 Output LM 741 dari Sensor Arus Pada Kamar 1



Gambar 4.10 Output XOR Pada Kamar 1

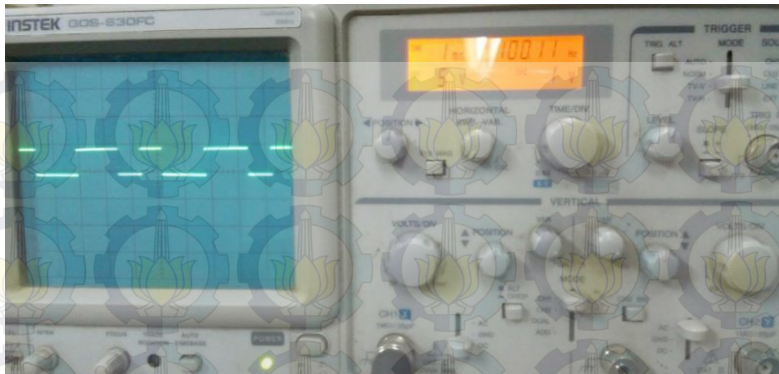
Gelombang yang dilihat dari osiloskop berasal dari *power transformer* kamar 1 pada Gambar 4.8, *current transformer* kamar 1 pada Gambar 4.9, dan *output IC XOR* kamar 1 pada Gambar 4.10, dapat dilihat bahwa ke-tiga gelombang mempunyai besaran 0-5 V, itu menunjukkan bahwa gelombang tersebut dapat di proses dalam interrupt mikrokontroler. Kemudian untuk gelombang yang dihasilkan oleh IC XOR kamar 1 adalah selisih waktu antara gelombang dari sensor tegangan kamar 1 dengan gelombang sensor arus kamar 1. *Output* dari IC XOR kamar 1 inilah yang akan diproses di interrupt mikrokontroler.



Gambar 4.11 Output LM 741 dari Sensor Tegangan Pada Kamar 2



Gambar 4.12 Output LM 741 dari Sensor Arus Pada Kamar 2



Gambar 4.13 Output XOR Pada Kamar 2

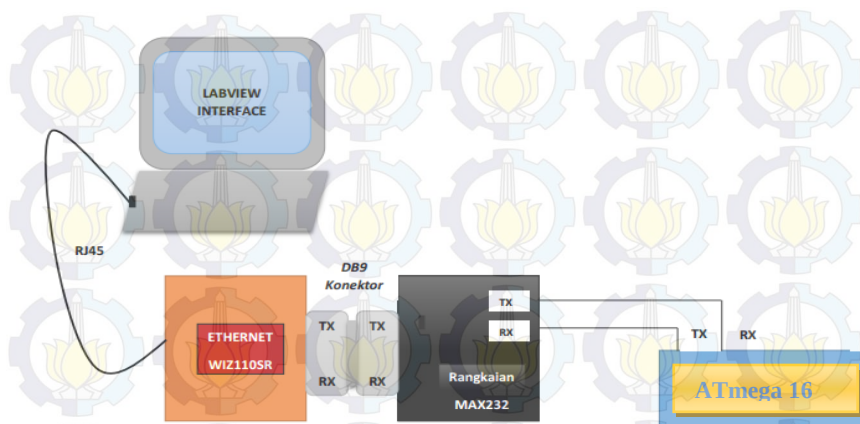
Tabel 4.8 Data Pengambilan $\cos\phi$

No	Beban	Cosphi meter	Sensor Cosphi	Persentase Eror
1	Lampu 110 Watt	1	0,98	2 persen
2	Setrika	1	1	0 persen
3	Kipas Angin Duduk	0,65	0,6	7,8 persen

Gelombang yang dilihat dari osiloskop berasal dari *power transformer* kamar 2 pada Gambar 4.11, *current transformer* kamar 2 pada Gambar 4.12, dan *output IC XOR* kamar 2 pada Gambar 4.13, dapat dilihat bahwa ke-tiga gelombang mempunyai besaran 0-5 V, itu menunjukkan bahwa gelombang tersebut dapat di proses dalam *interrupt* mikrokontroler. Kemudian untuk gelombang yang dihasilkan oleh IC XOR kamar 2 adalah selisih waktu antara gelombang dari sensor tegangan kamar 2 dengan gelombang sensor arus kamar 2. *Output* dari IC XOR kamar 2 inilah yang akan diproses di *interrupt* mikrokontroler.

Dari data $\cos\phi$ yang di dapat pada Tabel 4.8, jenis beban mempengaruhi besar persentase eror yang muncul. Pada beban Kipas Angin mengalami persentase eror yang besar, sedangkan pada beban lainnya persentase eror kecil. Sensor $\cos\phi$ ini dapat digunakan karena persentase eror yang muncul tidak besar.

4.6 Pengujian Rangkaian Komunikasi



Gambar 4.14 Skema Rangkaian Komunikasi

Pada Tugas Akhir ini menggunakan *serial komunikasi Ethernet* yang dihubungkan dengan PC Komputer dengan menggunakan kabel komunikasi serial RJ45. Untuk skema rangkaian komunikasi dapat dilihat pada Gambar 4.14 dibawah ini. Kemudian untuk pengujian koneksi rangkaian komunikasi menggunakan Rangkaian MAX232 dan *Setting IP Rangkaian Ethernet WIZ110SR*.

4.6.1 Pengujian Rangkaian MAX232

Bila *jack female* pada rangkaian MAX232 disambungkan ke rangkaian Ethernet WIZ110SR dengan menggunakan konektor DB9. Setelah itu baru bisa dilakukan pengujian terhadap rangkaian MAX232. Langkah-langkah pengujian dijelaskan sebagai berikut :

1. Buka *hyperterminal*
Start => Programs => Accesories => Communication => pilih Hyper Terminal. Setelah itu akan muncul tampilan seperti Gambar 4.6.



Gambar 4.15 Tampilan *Hyper Terminal*



Gambar 4.16 Tampilan *Testing MAX232*

2. Beri nama (misal = tes) kemudian klik OK maka akan muncul tampilan seperti Gambar 4.16.
3. Kemudian pilih *port* serial yang akan digunakan, kemudian klik ok maka akan muncul jendela seperti Gambar 4.16.



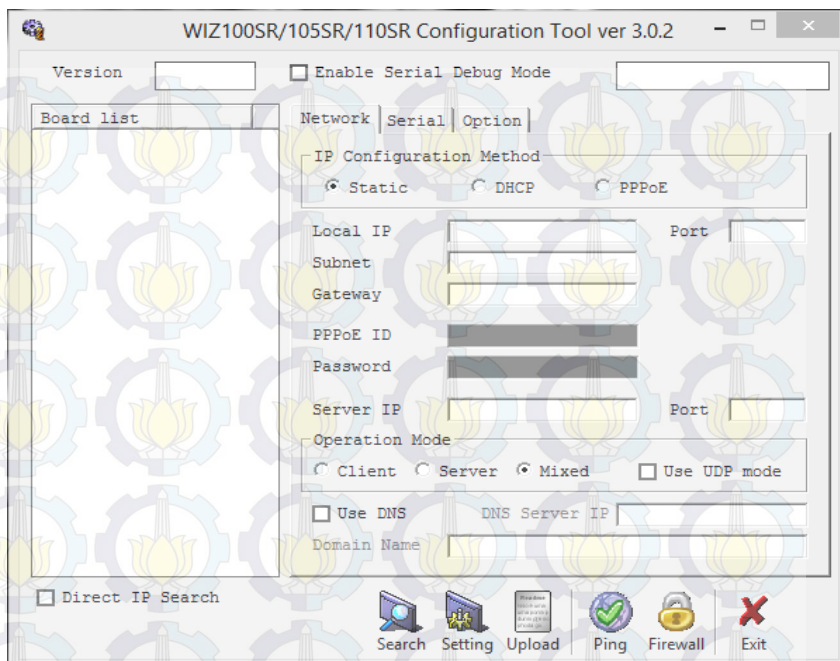
Gambar 4.17 Tampilan *Port Setting*

4. Aturlah seperti pengaturan pada Gambar 4.17 diatas lalu klik ok. Maka data yang dikirimkan oleh Mikrokontroler akan ditampilkan pada *hyper terminal*.

4.6.2 Setting IP Rangkaian Ethernet WIZ110SR

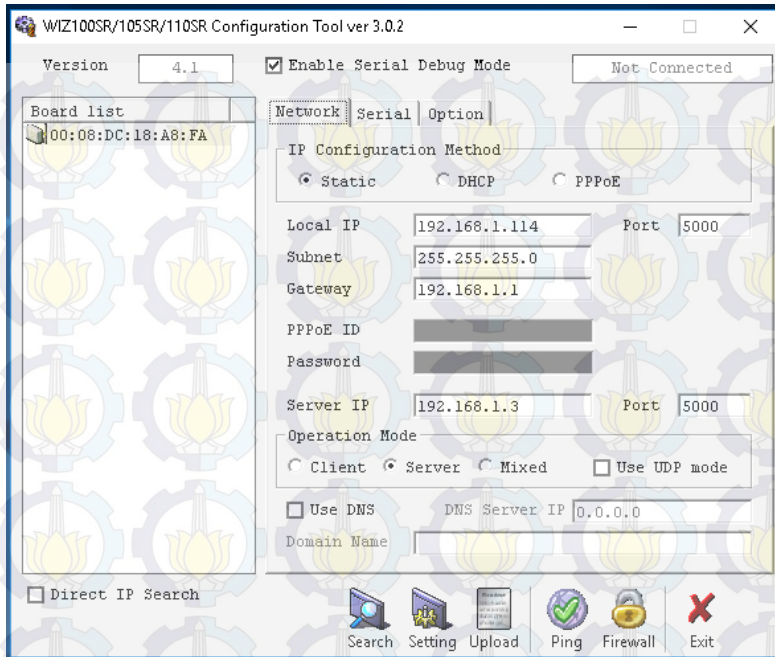
xUntuk dapat mengakses Ethernet WIZ110SR ke PC, terlebih dahulu melakukan penyettingan *IP address*. Langkah-langkah untuk penyettingan pada Ethernet WIZ110SR adalah sebagai berikut :

1. Koneksikan terlebih dahulu antara PC dan Rangkaian Ethernet WIZ110SR dengan menggunakan kabel RJ45.
2. Menginstal software WIZ10XSR *Configuration Tool*.
3. Buka software WIZ10XSR *Configuration Tool* sehingga muncul jendela seperti pada Gambar 4.9.

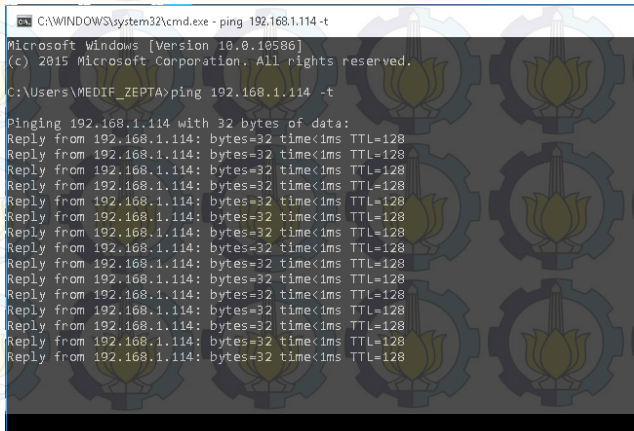


Gambar 4.18 Tampilan WIZ10XSr Configuration Tool

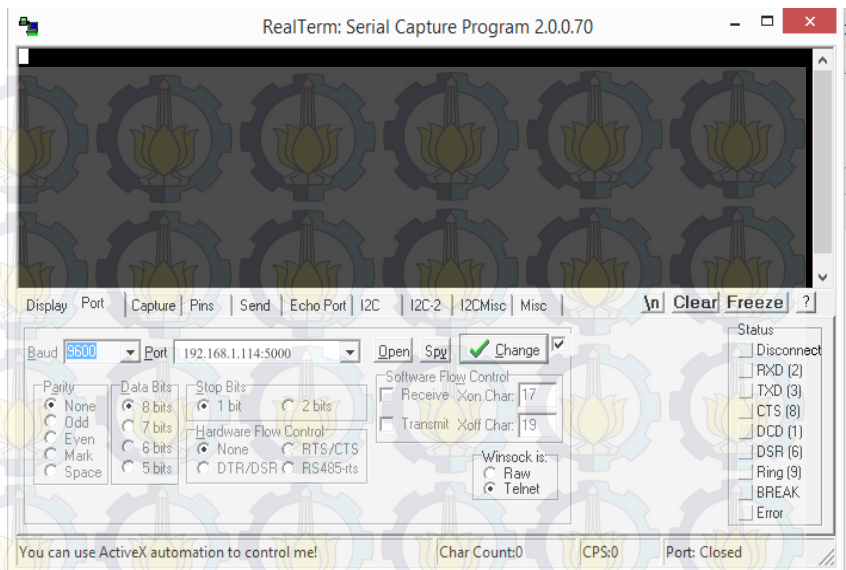
4. Lalu isikan Serial IP Address yang diinginkan seperti Gambar 4.19, pastikan IP address yang diisikan sesuai agar dapat terkoneksi.
5. Setelah itu simpan settingan IP yang telah diisikan.
6. Untuk mengetes koneksi apakah antara PC dan Ethernet telah tersambung maka harus dilakukan Testing yaitu dengan mengirimkan “ping.” untuk memanggil Ethernet seperti pada Gambar 4.20.



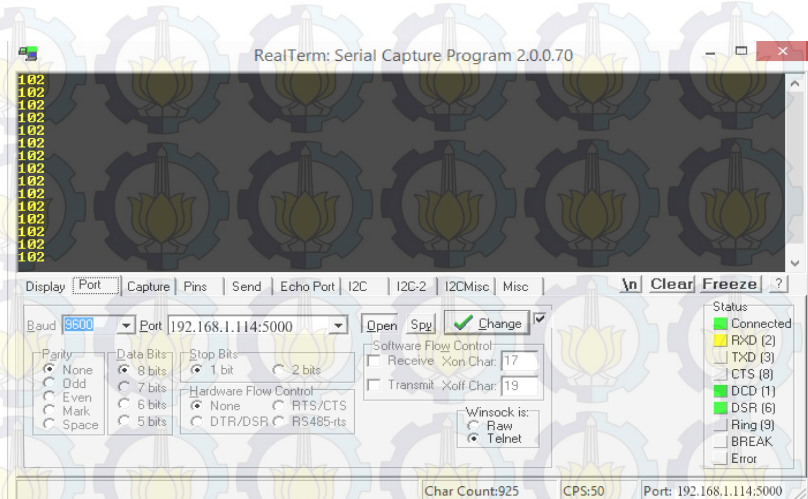
Gambar 4.19 Pengisian IP Address Ethernet WIZ110SR



Gambar 4.20 Pengiriman Data untuk Memanggil Router



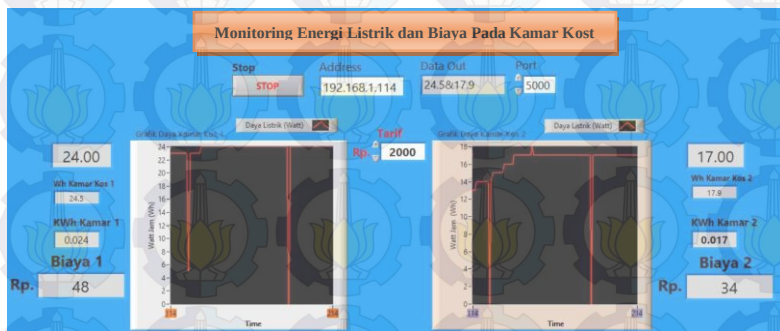
Gambar 4.21 Tampilan Software RealTerm



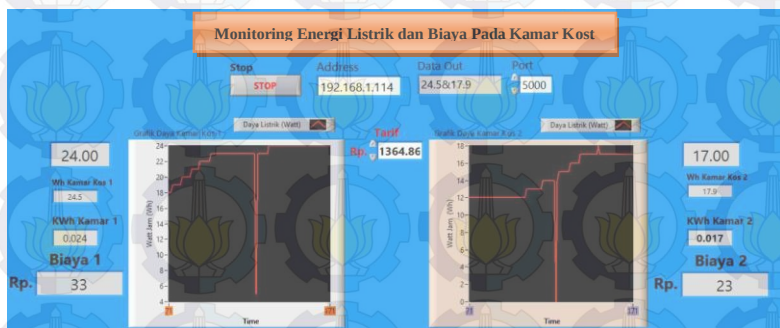
Gambar 4.22 Penerimaan Data Pada RealTerm

7. Setelah mendapat muncul tulisan seperti Gambar 4.21 maka artinya telah terkoneksi dan siap melakukan pengiriman data.
8. Lalu setelah itu buka *software RealTerm* untuk percobaan pengiriman data.
9. Lalu isikan kembali *Ip Address* serta *BaudRate*.
10. Setelah itu akan muncul indikasi ketika berhasil terkoneksi, dan komputer akan menerima data. Untuk tampilannya bisa dilihat pada Gambar 4.22.

4.6.3 Pengujian LabVIEW



Gambar 4.23 Tampilan LabVIEW Dengan Tarif 2000 per kWh



Gambar 4.24 Tampilan LabVIEW Dengan Tarif 1264.86 per kWh

Pada Gambar 4.23 dan Gambar 4.24 menunjukkan kondisi saat dimana penetapan tarif berubah. Namun pada perubahan nilai tarif tersebut, biaya yang kondisi sebelum tarif berubah akan hilang dan berganti dengan biaya tarif yang baru. Hal ini dikarenakan LabVIEW tersebut belum dilengkapi fitur penyimpanan data internal pada kolom Biaya 1 maupun Biaya 2.

4.7 Pengujian Beban Variasi

Setelah pengujian alat secara keseluruhan maka dilakukan pengujian beban bervariasi, pada pengujian ini terdapat lima variabel yaitu lampu pijar 100 Watt (0,37 A), *Electril Drill* 400 Watt (0,38 A), *Charger Laptop* 65 Watt (0,06 A), Kipas Angin 45 Watt (0,13 A), Setrika 350 Watt (1,4 A). Dibawah ini adalah data yang diambil dari lima variabel diatas.

Rumus Perhitungan Daya :

$$P = V \times I \times \cos \varphi$$

$$V = 231,2 \text{ Volt}$$

Setiap sistem mempunyai nilai eror yang mempengaruhi kerja sistem. Untuk mengetahui tingkat eror sistem, dapat menggunakan rumus dibawah ini.

Rumus Perhitungan Eror:

$$\text{Persentase Eror} = 100\% - \left(\frac{\text{Daya Pengukuran}}{\text{Daya Perhitungan}} \times 100\% \right) \dots\dots(4.1)$$

Tabel 4.9 Data Penggunaan Daya Kamar 1

Beban	Daya Pengukuran	Daya Perhitungan	Persentase Eror
1 Lampu + Kipas Angin	121 Watt	115,6 Watt	4,6 persen
<i>Electril Drill</i> + 1 Lampu	176 Watt	173,4 Watt	1,49 persen
<i>Electril Drill</i> + Setrika	412 Watt	411,5 Watt	0,12 persen
Setrika + 2 Lampu	505 Watt	504,016 Watt	0,19 persen
<i>Charger Laptop</i> + 1 Lampu + <i>Electril Drill</i>	208 Watt	208,8 Watt	0,39 persen

Tabel 4.10 Data Penggunaan Daya Kamar 2

Beban	Daya Pengukuran	Daya Perhitungan	Persentase Eror
3 Lampu + <i>Charger Laptop</i>	302 Watt	312,12 Watt	3,2 persen
Kipas Angin + 2 Lampu + Setrika	544 Watt	534,072 Watt	1,85 persen
Kipas Angin + Setrika + <i>Electril Drill</i>	446 Watt	441,592 Watt	0,99 persen
<i>Electril Drill</i> + <i>Charger Laptop</i>	126 Watt	122,536 Watt	2,82 persen
1 Lampu + 1 <i>Charger Laptop</i>	121 Watt	120,224 Watt	0,64 persen

Pada Tabel 4.9 menunjukkan hasil pengukuran yang dihasilkan oleh alat pada kamar 1, kemudian dibandingkan dengan data perhitungan manual. Untuk mengetahui ke akuratan data yang dihasilkan alat pada kamar 1, dilakukan perbandingan hasil pengukuran pada kamar 1 dengan perhitungan dengan rumus persamaan 4.1. Kemudian hasil dari perbandingan tersebut dimasukkan dalam kolom persentase eror.

Pada kolom persentase eror, menunjukkan bahwa perbandingan antara pengukuran pada kamar 1 dengan perhitungan mempunyai persentase eror kurang dari 5 persen. Hasil tersebut menunjukkan bahwa alat pada kamar 1 mempunyai nilai keakuratan diatas 95 persen dan hasil alat dapat berfungsi dengan baik.

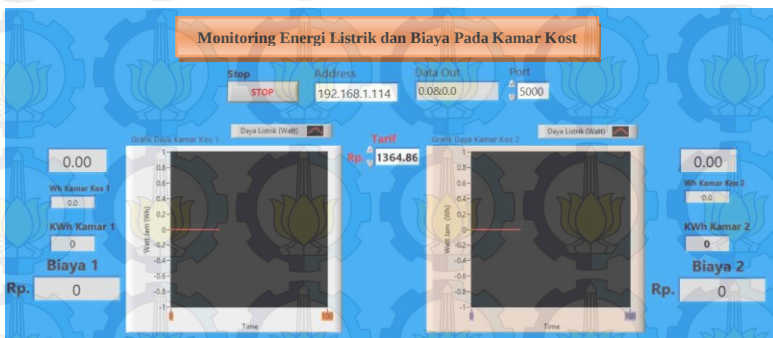
Pada Tabel 4.10 diatas menunjukkan hasil pengukuran yang dihasilkan oleh alat pada kamar 2, kemudian dibandingkan dengan data perhitungan manual. Untuk mengetahui ke akuratan data yang dihasilkan alat pada kamar 2, dilakukan berbandingan hasil pengukuran pada kamar 2 dengan perhitungan. Kemudian hasil dari perbandingan tersebut dimasukkan dalam kolom persentase eror.

Pada kolom persentase eror, menunjukkan bahwa perbandingan antara pengukuran pada kamar 2 dengan perhitungan mempunyai persentase eror kurang dari 5 persen. Hasil tersebut menunjukkan bahwa alat pada kamar 2 mempunyai nilai keakuratan diatas 95 persen dan hasil alat dapat berfungsi dengan baik.

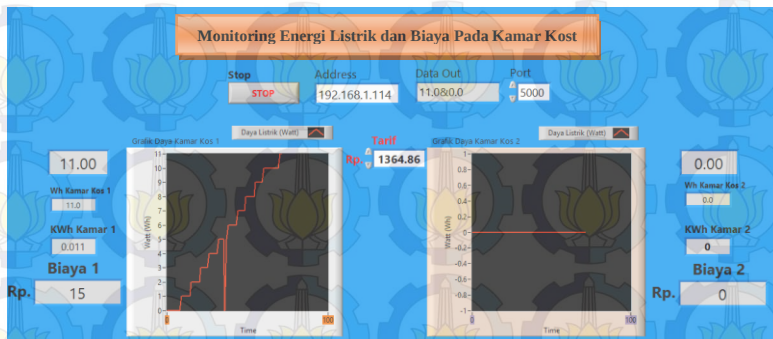
4.8 Pengujian Keseluruhan

Percobaan pertama, saat rangkaian tanpa beban dan di-monitoring dengan *LabVIEW* seperti pada Gambar 4.25. Pada kolom Wh kamar1 dan juga grafik menunjukkan nilai 0 yang menyatakan nilai daya terhadap waktu selama tanpa beban

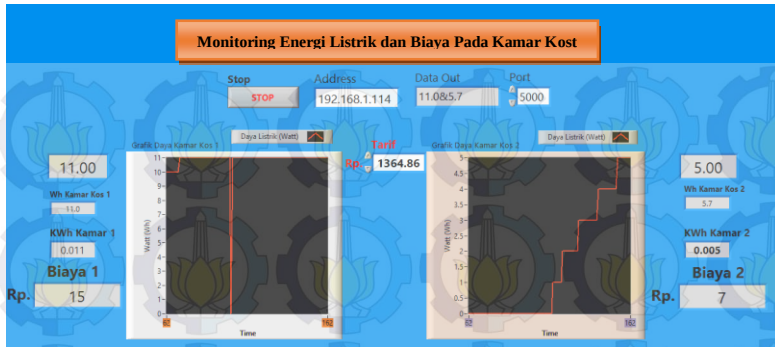
Pada percobaan kedua, menggunakan beban setrika pada kamar 1 dan tanpa beban pada kamar 2. Tampilan *LabVIEW* terlihat pada Gambar 4.26. Kolom Wh1 menunjukkan nilai 11,8, sedangkan kolom Wh2 menunjukkan nilai 0. Sedangkan nilai biaya adalah Rp. 15 yang merupakan hasil dari perkalian antara kWh dengan dengan tarif.



Gambar 4.25 Tampilan LabVIEW Tanpa Beban



Gambar 4.26 Tampilan LabVIEW Pada Kamar 1



Gambar 4.27 Tampilan LabVIEW Pada Kamar 2

Pada percobaan ketiga, beban pada kamar 1 dicabut, sehingga kamar 1 tanpa beban, pada kamar 2 menggunakan setrika sebagai beban. Terlihat pada Gambar 4.24. Kolom Wh1 menunjukkan nilai 11,8, sedangkan kolom Wh2 menunjukkan nilai 5,7. Sedangkan nilai biaya 1 adalah Rp. 15 dan nilai Biaya 2 adalah Rp. 7 yang merupakan hasil dari perkalian antara kWh pada setiap kamar dengan tarif.

DAFTAR PUSTAKA

- [1] Sukmawardi, I., Atnawijaya, H. B., "Rancang Bangun Alat Penghitung Penggunaan Daya Listrik Untuk Keperluan Kamar Kost Tersambung dengan Intranet", *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
- [2] Azid, M., Syaifullah, T., "Rancang Bangun Monitoring Pemakaian Daya Listrik Pada Miniatur Panel Listrik", *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2013.
- [3] Hayt, W. H., Kemmerly, J. E., Silaban, P., "*Rangkaian Listrik*", Erlangga, Jakarta, 1988.
- [4] Gates, E. D., "*Introdunction To Electronics*", Delmar, New York, 2012.
- [5] Taufan, Muhammad, *Current Transformator (CT) atau Trafo Arus*, <http://engineeringbuilding.blogspot.co.id/2011/08/current-transformer-ct-atau-trafo-arus.html/>, 21 Agustus 2013.
- [6] Kartinisari, E. N., Santoso, B. W., "Sistem Monitoring Daya dan Perancangan Switchung Kapasitor Untuk Perbaikan Faktor Daya Otomatis di Industri", *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2012.
- [7] Wardhana, L., "*Belajar Sendiri Mikrokontroler AVR Seri ATmega16 Simulasi, Hardware, dan Aplikasi*", ANDI, Yogyakarta, 2006.
- [8] _____, "*Planar UltraRes Series RS 232 User Manual*", Planar Sistems Inc., USA, 2015.
- [9] _____, "*WIZ110SR User's Manual (Version 1.0)*", WIZnet Ltd., Korea, 2007.
- [10] Sigiyo, Stevani Agnesia., *Instrumentasi Virtual Menggunakan Labview dan Soundcard*. <http://repository.usu.ac.id/handle/123456789/45580>, 16 April 2015.
- [11] K. Kondo, H. Terao, H. Abe, M. Ohta, K. Suzuki, T. Sasaki, G. Kawachi, J. Ohwada, "*Liquid Crystal Display Device*", filed Sep 18, 1992 and Jan 20, 1993.

BAB V PENUTUP

Setelah melakukan perencanaan dan pembuatan alat serta pengujian dan analisis, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan.

5.1 Kesimpulan

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

1. Alat penghitung penggunaan energi listrik dan biaya penggunaan listrik mampu bekerja dengan akurat.
2. Hasil dari pengambilan data menunjukkan persentase eror dibawah 5 persen yang menunjukkan bahwa alat bekerja dengan akurat.
3. Data yang dikirim ke *LabVIEW* terkadang mengalami eror, namun tidak berakibat fatal pada data biaya listrik.
4. Sistem *interface* yang berupa *LabVIEW* mempunyai kekurangan yaitu harus di pantau dengan laptop menyala, bila tidak data akan hilang.

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. Pengambilan data lebih banyak dan berulang, untuk memastikan kemampuan alat serta keakuratan alat.
2. Memakai perangkat elektronik yang lebih kuat dan tahan lama, serta saat pemasangan harus diperhatikan bahwa hasil solderan kuat.
3. Dalam *me-monitoring* data yang ada sebaiknya dilakukan secara bertahap dengan rentang waktu yang stabil.
4. Data penggunaan energi listrik sebaiknya di *upload* ke internet dan diakses lewat *website*, agar data yang disimpan tidak hilang dan tidak harus diakses menggunakan laptop.

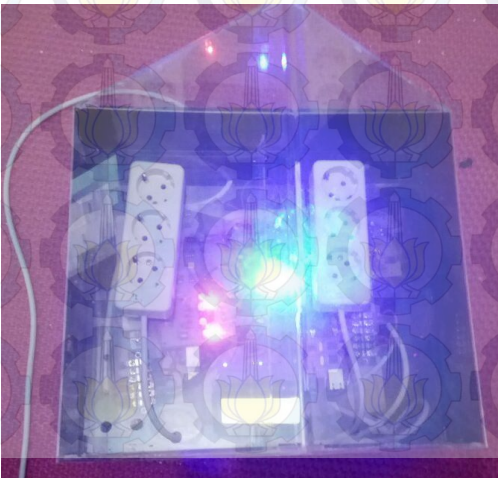
LAMPIRAN A

FOTO

1. Foto Tampak Depan



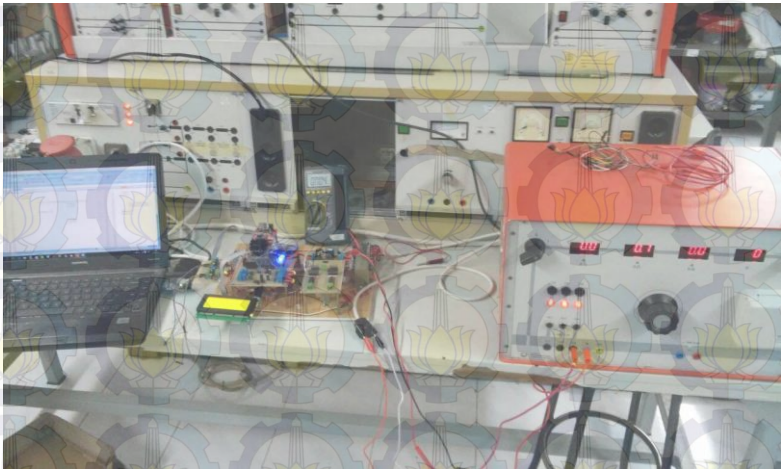
2. Foto Tampak Atas



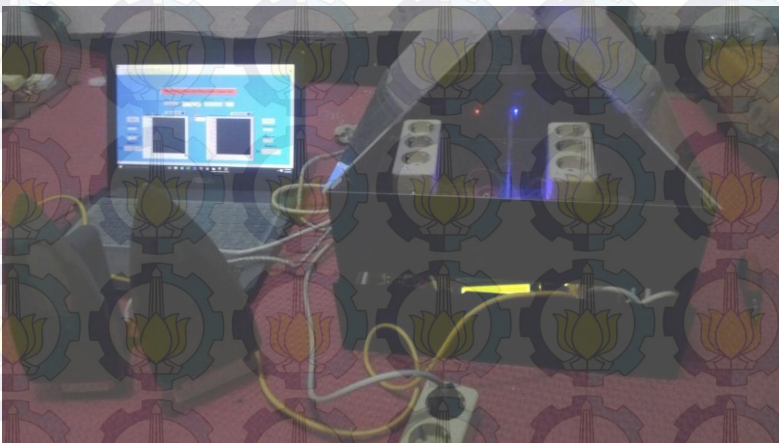
3. Foto Tampak Atas (Sliding)

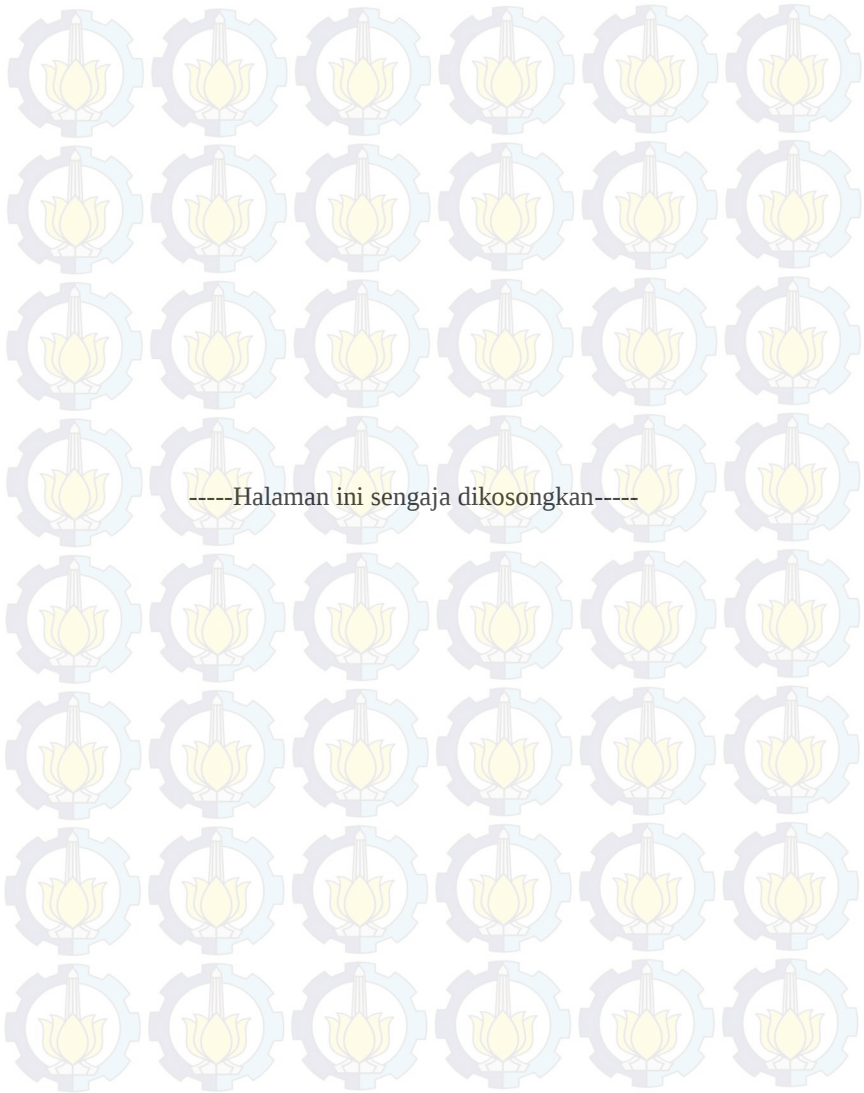


4. Foto Pengujian Keseluruhan



5. Foto Pengujian Keseluruhan Menggunakan LabVIEW





LAMPIRAN B PROGRAM

/*****

This program was created by the
CodeWizardAVR V3.12 Advanced
Automatic Program Generator

© Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project : Tugas Akhir

Version : 3.12

Date : 14/06/2016

Author 1: Medif Zepta Yogassasena (2213039008)

Author 2: Betty Nur Qamarina (2213039023)

Company : Institut Teknologi Sepuluh Nopember

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 8,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

#include <mega8535.h>

#include <stdio.h>

#include <stdlib.h>

#include <delay.h>

#include <string.h>

// Alphanumeric LCD functions

#include <alcd.h>

// Declare your global variables here

int adc0,adc1,adc2,adc3;

```

char buff[8];
char biff[8];
char baff[8];
int timer0 = 0;
int timer2 = 0;
int time0 = 0;
int time2 = 0;
float av1, av2, ai1, ai2, bv1, bi1, bv2, bi2,p1,p2,w1,w2,wh1,wh2;
float cosphi1 = 0;
float cosphi2 = 0;
char gabung[20];

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here
    if(timer0<150)
    { time0 = timer0 * 32; }
    timer0 = 0;
    cosphi1 = cos((time0*90)/5000);
}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    // Place your code here
    if(timer2<150)
    { time2 = timer2 * 32; }
    timer2 = 0;
    cosphi2 = cos((time2*90)/5000);
}

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

```

```

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

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index=0,rx_rd_index=0;
#else
unsigned int rx_wr_index=0,rx_rd_index=0;
#endif

#if RX_BUFFER_SIZE < 256
unsigned char rx_counter=0;
#else
unsigned int rx_counter=0;
#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_BUFFER_SIZE == 256
        // special case for receiver buffer size=256
        if (++rx_counter == 0) rx_buffer_overflow=1;
        #else
        if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
        if (++rx_counter == RX_BUFFER_SIZE)
        {
            rx_counter=0;
            rx_buffer_overflow=1;
        }
        #endif
    }
}

```

```

    }
#endif
}
}

#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_BUFFER_SIZE != 256
    if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #endif
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}
#pragma used-
#endif

// USART Transmitter buffer
#define TX_BUFFER_SIZE 8
char tx_buffer[TX_BUFFER_SIZE];

#if TX_BUFFER_SIZE <= 256
unsigned char tx_wr_index=0,tx_rd_index=0;
#else
unsigned int tx_wr_index=0,tx_rd_index=0;
#endif

#if TX_BUFFER_SIZE < 256
unsigned char tx_counter=0;
#else
unsigned int tx_counter=0;

```

#endif

```
// USART Transmitter interrupt service routine
interrupt [USART_TXC] void usart_tx_isr(void)
{
    if (tx_counter)
    {
        --tx_counter;
        UDR=tx_buffer[tx_rd_index++];
        #if TX_BUFFER_SIZE != 256
            if (tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
        #endif
    }
}

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
    while (tx_counter == TX_BUFFER_SIZE);
    #asm("cli")
    if (tx_counter || ((UCSRA & DATA_REGISTER_EMPTY)==0))
    {
        tx_buffer[tx_wr_index++]=c;
        #if TX_BUFFER_SIZE != 256
            if (tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
        #endif
        ++tx_counter;
    }
    else
        UDR=c;
    #asm("sei")
}
#pragma used-
#endif
```

// Timer 0 overflow interrupt service routine

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
```

```
// Place your code here
```

```
timer0++;
```

```
}
```

```
// Timer2 overflow interrupt service routine
```

```
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
```

```
{
```

```
// Place your code here
```

```
timer2++;
```

```
}
```

```
// Standard Input/Output functions
```

```
#include <stdio.h>
```

```
// Voltage Reference: AREF pin
```

```
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) |  
(0<<ADLAR))
```

```
// Read the AD conversion result
```

```
unsigned int read_adc(unsigned char adc_input)
```

```
{
```

```
ADMUX=adc_input | ADC_VREF_TYPE;
```

```
// Delay needed for the stabilization of the ADC input voltage
```

```
delay_us(10);
```

```
// Start the AD conversion
```

```
ADCSRA|=(1<<ADSC);
```

```
// Wait for the AD conversion to complete
```

```
while ((ADCSRA & (1<<ADIF))==0);
```

```
ADCSRA|=(1<<ADIF);
```

```
return ADCW;
```

```
}
```

```
void primer()
```

```
{
```

```
gabung[0]='\0';
```

```
av1=0.254438383;
```

```
bv1=12.10098663;
```

```

ai1=0.004473333;
bi1=0.397174395;
av2=0.256435616;
bv2=11.77439617;
ai2=0.004494327;
bi2=0.389961716;

adc0=read_adc(0);
v1=((av1*adc0)-bv1);

adc1=read_adc(1);
i1=((ai1*adc1)-bi1);

adc2=read_adc(2);
v2=((av2*adc2)-bv2);

adc3=read_adc(3);
i2=((ai2*adc3)-bi2);

p1=v1*i1*cosphi1;
p2=v2*i2*cosphi2;

w1=w1+p1;
wh1=((w1+p1)/3600);
w2=w2+p2;
wh2=((w2+p2)/3600);
lcd_clear();
delay_ms(1);

lcd_gotoxy(7,0);
lcd_puts("KAMAR 1");
lcd_gotoxy(0,1);
lcd_puts("Wh1=");
lcd_gotoxy(6,1);
ftoa(wh1,1,buff);
lcd_puts(buff);

```

```

    lcd_gotoxy(7,2);
    lcd_puts("KAMAR 2");
    lcd_gotoxy(0,3);
    lcd_puts("Wh2=");
    lcd_gotoxy(6,3);
    ftoa(wh2,1,biff);
    lcd_puts(biff);
    strcat(gabung,buff);
    strcat(gabung,"&");
    strcat(gabung,biff);

    delay_ms(1000);
}

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
    // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In
    Bit0=In
    DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) |
    (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
    // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
    PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) |
    (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) |
    (0<<PORTA0);

    // Port B initialization
    // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In
    Bit0=In
    DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) |
    (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
    // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
    PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) |
    (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) |
    (0<<PORTB0);

```

// Port C initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In
Bit0=In

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) |
(0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) |
(0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) |
(0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In
Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) |
(0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) |
(0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) |
(0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: 8000,000 kHz

// Mode: Normal top=0xFF

// OC0 output: Disconnected

// Timer Period: 0,032 ms

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (1<<CS00);

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected

// OC1B output: Disconnected

// 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=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) |
(0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) |
(0<<CS12) | (0<<CS11) | (0<<CS10);
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: 8000,000 kHz
// Mode: Normal top=0xFF
// OC2 output: Disconnected
// Timer Period: 0,032 ms
ASSR=0<<AS2;
TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) |
(0<<CS22) | (0<<CS21) | (1<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (1<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (1<<TOIE0);

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Any change
// INT1: On
// INT1 Mode: Any change

```

```

// INT2: Off
GICR=(1<<INT1) | (1<<INT0) | (0<<INT2);
MCUCR=(0<<ISC11) | (1<<ISC10) | (0<<ISC01) | (1<<ISC00);
MCUCSR=(0<<ISC2);
GIFR=(1<<INTF1) | (1<<INTF0) | (0<<INTF2);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) |
(0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(1<<RXCIE) | (1<<TXCIE) | (0<<UDRIE) | (1<<RXEN) |
(1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) |
(0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) |
(0<<ADIE) | (0<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);

```

```

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL)
| (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) |
(0<<TWIE);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(20);

// Global enable interrupts
#asm("sei")

while (1)
{
    // Place your code here

    primer();
    puts(gabung);
}
}

```

LAMPIRAN C

DATASHEET

1. Datasheet Mikrokontroler ATmega 16

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

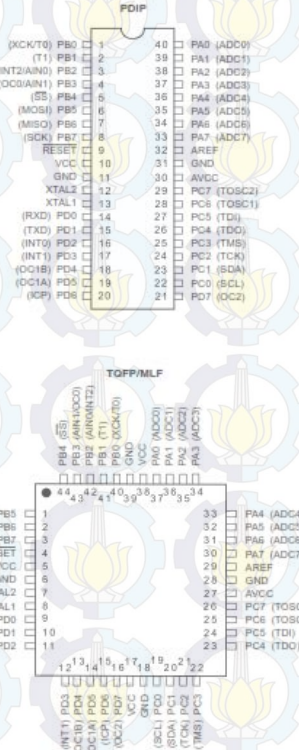
Rev. 2460E-AVR-10/02





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.

The EEPROM Address Register – EEARH and EEARL

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	–	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

• Bits 15.9 – Res: Reserved Bits

These bits are reserved bits in the ATmega16 and will always read as zero.

• Bits 8..0 – EEAR8..0: EEPROM Address

The EEPROM Address Registers – EEARH and EEARL – specify the EEPROM address in the 512 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 511. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

The EEPROM Data Register – EEDR

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7..0 – EEDR7..0: EEPROM Data

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

The EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

• Bits 7..4 – Res: Reserved Bits

These bits are reserved bits in the ATmega16 and will always read as zero.

• Bit 3 – EERIE: EEPROM Ready Interrupt Enable

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEWE is cleared.

• Bit 2 – EEMWE: EEPROM Master Write Enable

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set, setting EEWE within four clock cycles will write data to the EEPROM at the selected address. If EEMWE is zero, setting EEWE will have no effect.



Interrupts

Interrupt Vectors in ATmega16

This section describes the specifics of the interrupt handling as performed in ATmega16. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 11.

Table 18. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	\$000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$002	INT0	External Interrupt Request 0
3	\$004	INT1	External Interrupt Request 1
4	\$006	TIMER2 COMP	Timer/Counter2 Compare Match
5	\$008	TIMER2 OVF	Timer/Counter2 Overflow
6	\$00A	TIMER1 CAPT	Timer/Counter1 Capture Event
7	\$00C	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	\$00E	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	\$010	TIMER1 OVF	Timer/Counter1 Overflow
10	\$012	TIMER0 OVF	Timer/Counter0 Overflow
11	\$014	SPI, STC	Serial Transfer Complete
12	\$016	USART, RXC	USART, Rx Complete
13	\$018	USART, UDRE	USART Data Register Empty
14	\$01A	USART, TXC	USART, Tx Complete
15	\$01C	ADC	ADC Conversion Complete
16	\$01E	EE_RDY	EEPROM Ready
17	\$020	ANA_COMP	Analog Comparator
18	\$022	TWI	Two-wire Serial Interface
19	\$024	INT2	External Interrupt Request 2
20	\$026	TIMER0 COMP	Timer/Counter0 Compare Match
21	\$028	SPM_RDY	Store Program Memory Ready

- Notes:
1. When the BOOTRST fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 241.
 2. When the IVSEL bit in GICR is set, interrupt vectors will be moved to the start of the Boot Flash section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash section.

Table 19 shows Reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 19. Reset and Interrupt Vectors Placement⁽¹⁾

BOOTRST	IVSEL	Reset address	Interrupt Vectors Start Address
1	0	\$0000	\$0002
1	1	\$0000	Boot Reset Address + \$0002
0	0	Boot Reset Address	\$0002
0	1	Boot Reset Address	Boot Reset Address + \$0002

Note: 1. The Boot Reset Address is shown in Table 99 on page 252. For the BOOTRST Fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega16 is:

Address	Labels	Code	Comments
\$000		<code>jmp RESET</code>	; Reset Handler
\$002		<code>jmp EXT_INT0</code>	; IRQ0 Handler
\$004		<code>jmp EXT_INT1</code>	; IRQ1 Handler
\$006		<code>jmp TIM2_COMP</code>	; Timer2 Compare Handler
\$008		<code>jmp TIM2_OVF</code>	; Timer2 Overflow Handler
\$00A		<code>jmp TIM1_CAPT</code>	; Timer1 Capture Handler
\$00C		<code>jmp TIM1_COMPB</code>	; Timer1 CompareB Handler
\$00E		<code>jmp TIM1_COMPA</code>	; Timer1 CompareA Handler
\$010		<code>jmp TIM1_OVF</code>	; Timer1 Overflow Handler
\$012		<code>jmp TIM0_OVF</code>	; Timer0 Overflow Handler
\$014		<code>jmp SPI_STC</code>	; SPI Transfer Complete Handler
\$016		<code>jmp UCART_RXC</code>	; UCART RX Complete Handler
\$018		<code>jmp UCART_UDRRE</code>	; UDR Empty Handler
\$01A		<code>jmp UCART_TXC</code>	; UCART TX Complete Handler
\$01C		<code>jmp ADC</code>	; ADC Conversion Complete Handler
\$01E		<code>jmp EE_RDY</code>	; EEPROM Ready Handler
\$020		<code>jmp ANA_COMP</code>	; Analog Comparator Handler
\$022		<code>jmp TWI2</code>	; Two-wire Serial Interface Handler
\$024		<code>jmp EXT_INT2</code>	; IRQ2 Handler
\$026		<code>jmp TIM0_COMP</code>	; Timer0 Compare Handler
\$028		<code>jmp SWR_RDY</code>	; Store Program Memory Ready Handler
;			
\$02A	RESET:	<code>ldi r16,high(RAMEND)</code>	; Main program start
\$02B		<code>out SPH,r16</code>	; Set stack pointer to top of RAM
\$02C		<code>ldi r16,low(RAMEND)</code>	
\$02D		<code>out SPL,r16</code>	
\$02E		<code>sei</code>	; Enable interrupts
\$02F		<code>cjmp \$,0x00</code>	
...	...	...	



Special Function I/O Register – SFIOR

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	ADHSM	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 2 – PUD: Pull-up disable

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ((DDxn, PORTxn) = 0b01). See “Configuring the Pin” on page 48 for more details about this feature.

Alternate Functions of Port A

Port A has an alternate function as analog input for the ADC as shown in Table 22. If some Port A pins are configured as outputs, it is essential that these do not switch when a conversion is in progress. This might corrupt the result of the conversion.

Table 22. Port A Pins Alternate Functions

Port Pin	Alternate Function
PA7	ADC7 (ADC input channel 7)
PA6	ADC6 (ADC input channel 6)
PA5	ADC5 (ADC input channel 5)
PA4	ADC4 (ADC input channel 4)
PA3	ADC3 (ADC input channel 3)
PA2	ADC2 (ADC input channel 2)
PA1	ADC1 (ADC input channel 1)
PA0	ADC0 (ADC input channel 0)

Table 23 and Table 24 relate the alternate functions of Port A to the overriding signals shown in Figure 26 on page 52.

Table 23. Overriding Signals for Alternate Functions in PA7..PA4

Signal Name	PA7/ADC7	PA6/ADC6	PA5/ADC5	PA4/ADC4
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	ADC7 INPUT	ADC6 INPUT	ADC5 INPUT	ADC4 INPUT

Table 24. Overriding Signals for Alternate Functions in PA3..PA0

Signal Name	PA3/ADC3	PA2/ADC2	PA1/ADC1	PA0/ADC0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 25.

Table 25. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	SS (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)

The alternate pin configuration is as follows:

• SCK – Port B, Bit 7

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB7. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB7. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB7 bit.

• MISO – Port B, Bit 6

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input regardless of the setting of DDB6. When the SPI is enabled as a Slave, the data direction of this pin is controlled by

DDRB6. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB6 bit.

▪ **MOSI – Port B, Bit 5**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDRB5. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDRB5. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB5 bit.

▪ **SS – Port B, Bit 4**

SS: Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDRB4. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDRB4. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB4 bit.

▪ **AIN1/OC0 – Port B, Bit 3**

AIN1: Analog Comparator Negative Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the analog comparator.

OC0: Output Compare Match output: The PB3 pin can serve as an external output for the Timer/Counter0 Compare Match. The PB3 pin has to be configured as an output (DDRB3 set (one)) to serve this function. The OC0 pin is also the output pin for the PWM mode timer function.

▪ **AIN0/INT2 – Port B, Bit 2**

AIN0: Analog Comparator Positive input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

INT2: External Interrupt Source 2: The PB2 pin can serve as an external interrupt source to the MCU.

▪ **T1 – Port B, Bit 1**

T1: Timer/Counter1 Counter Source.

▪ **T0/XCK – Port B, Bit 0**

T0: Timer/Counter0 Counter Source.

XCK: USART External Clock. The Data Direction Register (DDRB0) controls whether the clock is output (DDRB0 set) or input (DDRB0 cleared). The XCK pin is active only when the USART operates in Synchronous mode.

Table 26 and Table 27 relate the alternate functions of Port B to the overriding signals shown in Figure 26 on page 52. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 26. Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/SCK	PB6/MISO	PB5/MOSI	PB4/SS
PUOE	SPE • MSTR	SPE • MSTR	SPE • MSTR	SPE • MSTR
PUOV	PORTB7 • PUD	PORTB6 • PUD	PORTB5 • PUD	PORTB4 • PUD
DDOE	SPE • MSTR	SPE • MSTR	SPE • MSTR	SPE • MSTR
DDOV	0	0	0	0
PVOE	SPE • MSTR	SPE • MSTR	SPE • MSTR	0
PVOV	SCK OUTPUT	SPI SLAVE OUTPUT	SPI MSTR OUTPUT	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	SCK INPUT	SPI MSTR INPUT	SPI SLAVE INPUT	SPI SS
AIO	–	–	–	–

Table 27. Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/OC0/AIN1	PB2/INT2/AIN0	PB1/T1	PB0/T0/XCK
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC0 ENABLE	0	0	UMSEL
PVOV	OC0	0	0	XCK OUTPUT
DIEOE	0	INT2 ENABLE	0	0
DIEOV	0	1	0	0
DI	–	INT2 INPUT	T1 INPUT	XCK INPUT/T0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	–	–



Alternate Functions of Port C

The Port C pins with alternate functions are shown in Table 28. 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.

Table 28. Port C Pins Alternate Functions

Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin 2)
PC6	TOSC1 (Timer Oscillator Pin 1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

The alternate pin configuration is as follows:

• TOSC2 – Port C, Bit 7

TOSC2, Timer Oscillator pin 2: When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PC7 is disconnected from the port, and becomes the inverting output of the Oscillator amplifier. In this mode, a Crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

• TOSC1 – Port C, Bit 6

TOSC1, Timer Oscillator pin 1: When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PC6 is disconnected from the port, and becomes the input of the inverting Oscillator amplifier. In this mode, a Crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

• TDI – Port C, Bit 5

TDI, JTAG Test Data In: Serial input data to be shifted in to the Instruction Register or Data Register (scan chains). When the JTAG interface is enabled, this pin can not be used as an I/O pin.

• TDO – Port C, Bit 4

TDO, JTAG Test Data Out: Serial output data from Instruction register or Data Register. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

• TMS – Port C, Bit 3

TMS, JTAG Test Mode Select: This pin is used for navigating through the TAP-controller state machine. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

• TCK – Port C, Bit 2

TCK, JTAG Test Clock: JTAG operation is synchronous to TCK. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

• **SDA – Port C, Bit 1**

SDA, Two-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC1 is disconnected from the port and becomes the Serial Data I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation. When this pin is used by the Two-wire Serial Interface, the pull-up can still be controlled by the PORTC1 bit.

• **SCL – Port C, Bit 0**

SCL, Two-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC0 is disconnected from the port and becomes the Serial Clock I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation. When this pin is used by the Two-wire Serial Interface, the pull-up can still be controlled by the PORTC0 bit.

Table 29 and Table 30 relate the alternate functions of Port C to the overriding signals shown in Figure 26 on page 52.

Table 29. Overriding Signals for Alternate Functions in PC7..PC4

Signal Name	PC7/TOSC2	PC6/TOSC1	PC5/TDI	PC4/TDO
PUOE	AS2	AS2	JTAGEN	JTAGEN
PUOV	0	0	1	0
DDOE	AS2	AS2	JTAGEN	JTAGEN
DDOV	0	0	0	SHIFT_IR + SHIFT_DR
PVOE	0	0	0	JTAGEN
PVOV	0	0	0	TDO
DIOE	AS2	AS2	JTAGEN	JTAGEN
DIOV	0	0	0	0
DI	–	–	–	–
AIO	T/C2 OSC OUTPUT	T/C2 OSC INPUT	TDI	–

Table 30. Overriding Signals for Alternate Functions in PC3..PC0⁽¹⁾

Signal Name	PC3/TMS	PC2/TCK	PC1/SDA	PC0/SCL
PUOE	JTAGEN	JTAGEN	TWEN	TWEN
PUOV	1	1	PORTC1 • PUD	PORTC0 • PUD
DDOE	JTAGEN	JTAGEN	TWEN	TWEN
DDOV	0	0	SDA_OUT	SCL_OUT
PVOE	0	0	TWEN	TWEN
PVOV	0	0	0	0
DIOE	JTAGEN	JTAGEN	0	0
DIOV	0	0	0	0
DI	–	–	–	–
AIO	TMS	TCK	SDA INPUT	SCL INPUT

Note: 1. When enabled, the Two-wire Serial Interface enables slew-rate controls on the output pins PC0 and PC1. This is not shown in the figure. In addition, spike filters are connected between the AIO outputs shown in the port figure and the digital logic of the TWI module.

Alternate Functions of Port D

The Port D pins with alternate functions are shown in Table 31.

Table 31. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

The alternate pin configuration is as follows:

• OC2 – Port D, Bit 7

OC2, Timer/Counter2 Output Compare Match output: The PD7 pin can serve as an external output for the Timer/Counter2 Output Compare. The pin has to be configured as an output (DDD7 set (one)) to serve this function. The OC2 pin is also the output pin for the PWM mode timer function.

• ICP – Port D, Bit 6

ICP – Input Capture Pin: The PD6 pin can act as an Input Capture pin for Timer/Counter1.

• OC1A – Port D, Bit 5

OC1A, Output Compare Match A output: The PD5 pin can serve as an external output for the Timer/Counter1 Output Compare A. The pin has to be configured as an output (DDD5 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

• OC1B – Port D, Bit 4

OC1B, Output Compare Match B output: The PD4 pin can serve as an external output for the Timer/Counter1 Output Compare B. The pin has to be configured as an output (DDD4 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

• INT1 – Port D, Bit 3

INT1, External Interrupt Source 1: The PD3 pin can serve as an external interrupt source.

• INT0 – Port D, Bit 2

INT0, External Interrupt Source 0: The PD2 pin can serve as an external interrupt source.

• TXD – Port D, Bit 1

TXD, Transmit Data (Data output pin for the USART). When the USART Transmitter is enabled, this pin is configured as an output regardless of the value of DDD1.

• RXD – Port D, Bit 0

RXD, Receive Data (Data input pin for the USART). When the USART Receiver is enabled this pin is configured as an input regardless of the value of DDD0. When the USART forces this pin to be an input, the pull-up can still be controlled by the PORTD0 bit.

Table 32 and Table 33 relate the alternate functions of Port D to the overriding signals shown in Figure 26 on page 52.

Table 32. Overriding Signals for Alternate Functions PD7..PD4

Signal Name	PD7/OC2	PD6/ICP	PD5/OC1A	PD4/OC1B
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC2 ENABLE	0	OC1A ENABLE	OC1B ENABLE
PVOV	OC2	0	OC1A	OC1B
DICE	0	0	0	0
DIEOV	0	0	0	0
DI	–	ICP INPUT	–	–
AIO	–	–	–	–

Table 33. Overriding Signals for Alternate Functions in PD3..PDO

Signal Name	PD3/INT1	PD2/INT0	PD1/TXD	PD0/RXD
PUOE	0	0	TXEN	RXEN
PUOV	0	0	0	PORTD0 • PUD
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	0	0	TXEN	0
PVOV	0	0	TXD	0
DIOE	INT1 ENABLE	INT0 ENABLE	0	0
DIOV	1	1	0	0
DI	INT1 INPUT	INT0 INPUT	–	RXD
AIO	–	–	–	–

Register Description for I/O Ports

Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Input Pins Address – PINA

Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The double buffered Output Compare Register (OCR0) is compared with the Timer/Counter value at all times. The result of the compare can be used by the waveform generator to generate a PWM or variable frequency output on the Output Compare Pin (OC0). See “Output Compare Unit” on page 69. for details. The compare match event will also set the Compare Flag (OCF0) which can be used to generate an output compare interrupt request.

Definitions

Many register and bit references in this document are written in general form. A lower case “n” replaces the Timer/Counter number, in this case 0. However, when using the register or bit defines in a program, the precise form must be used i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions in Table 37 are also used extensively throughout the document.

Table 37. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0 register. The assignment is dependent on the mode of operation.

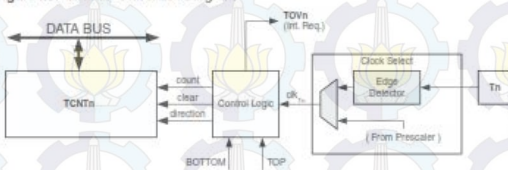
Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the clock select logic which is controlled by the clock select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0). For details on clock sources and prescaler, see “Timer/Counter0 and Timer/Counter1 Prescalers” on page 81.

Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 28 shows a block diagram of the counter and its surroundings.

Figure 28. Counter Unit Block Diagram



Signal description (internal signals):

- count** Increment or decrement TCNT0 by 1.
- direction** Select between increment and decrement.
- clear** Clear TCNT0 (set all bits to zero).
- clk_{Tn}** Timer/Counter clock, referred to as clk_{T0} in the following.
- TOP** Signalize that TCNT0 has reached maximum value.

Output Compare Unit

BOTTOM Signalize that TCNT0 has reached minimum value (zero).

Depending of the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T0}). clk_{T0} can be generated from an external or internal clock source, selected by the Clock Select bits (CS02:0). When no clock source is selected (CS02:0 = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of whether clk_{T0} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

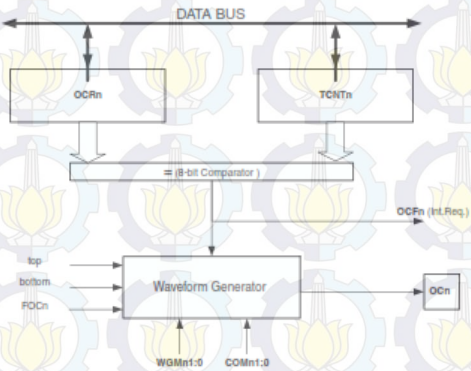
The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare output OC0. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 71.

The Timer/Counter Overflow (TOV0) flag is set according to the mode of operation selected by the WGM01:0 bits. TOV0 can be used for generating a CPU interrupt.

The 8-bit comparator continuously compares TCNT0 with the Output Compare Register (OCR0). Whenever TCNT0 equals OCR0, the comparator signals a match. A match will set the Output Compare Flag (OCF0) at the next timer clock cycle. If enabled (OCIE0 = 1 and Global Interrupt Flag in SREG is set), the Output Compare Flag generates an output compare interrupt. The OCF0 flag is automatically cleared when the interrupt is executed. Alternatively, the OCF0 flag can be cleared by software by writing a logical one to its I/O bit location. The waveform generator uses the match signal to generate an output according to operating mode set by the WGM01:0 bits and Compare Output mode (COM01:0) bits. The max and bottom signals are used by the waveform generator for handling the special cases of the extreme values in some modes of operation (See "Modes of Operation" on page 71).

Figure 29 shows a block diagram of the output compare unit.

Figure 29. Output Compare Unit. Block Diagram



The OCR0 Register is double buffered when using any of the Pulse Width Modulation (PWM) modes. For the normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR0 Compare Register to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR0 Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR0 Buffer Register, and if double buffering is disabled the CPU will access the OCR0 directly.

Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC0) bit. Forcing compare match will not set the OCF0 flag or reload/clear the timer, but the OC0 pin will be updated as if a real compare match had occurred (the COM01:0 bits settings define whether the OC0 pin is set, cleared or toggled).

Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 Register will block any compare match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0 to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT0 when using the output compare channel, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0 value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is downcounting.

The setup of the OC0 should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC0 value is to use the Force Output Compare (FOC0) strobe bits in Normal mode. The OC0 Register keeps its value even when changing between waveform generation modes.

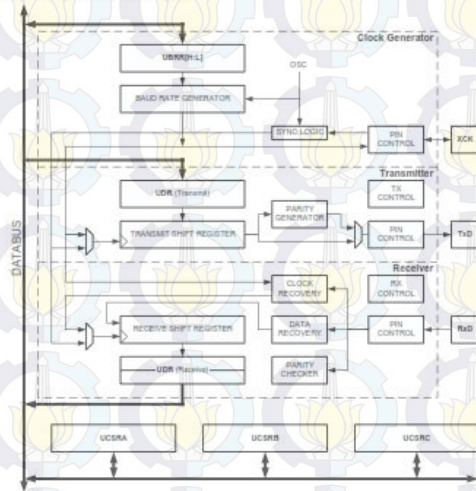
Be aware that the COM01:0 bits are not double buffered together with the compare value. Changing the COM01:0 bits will take effect immediately.

Compare Match Output Unit

The Compare Output mode (COM01:0) bits have two functions. The Waveform Generator uses the COM01:0 bits for defining the Output Compare (OC0) state at the next compare match. Also, the COM01:0 bits control the OC0 pin output source. Figure 30 shows a simplified schematic of the logic affected by the COM01:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port Control Registers (DDR and PORT) that are affected by the COM01:0 bits are shown. When referring to the OC0 state, the reference is for the internal OC0 Register, not the OC0 pin. If a System Reset occur, the OC0 Register is reset to "0".

- The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are:
 - Full Duplex Operation (Independent Serial Receive and Transmit Registers)
 - Asynchronous or Synchronous Operation
 - Master or Slave Clocked Synchronous Operation
 - High Resolution Baud Rate Generator
 - Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
 - Odd or Even Parity Generation and Parity Check Supported by Hardware
 - Data OverRun Detection
 - Framing Error Detection
 - Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
 - Three Separate Interrupts on TX Complete, TX Data Register Empty, and RX Complete
 - Multi-processor Communication Mode
 - Double Speed Asynchronous Communication Mode

Figure 69. USART Block Diagram⁽¹⁾



2466E-AVR-10/02



The dashed boxes in the block diagram separate the three main parts of the USART (listed from the top): Clock Generator, Transmitter and Receiver. Control registers are shared by all units. The clock generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator. The XCK (Transfer Clock) pin is only used by Synchronous Transfer mode. The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery units are used for asynchronous data reception. In addition to the recovery units, the receiver includes a parity checker, control logic, a Shift Register and a two level receive buffer (UDR). The receiver supports the same frame formats as the transmitter, and can detect frame error, data overrun and parity errors.

AVR USART vs. AVR UART – Compatibility

The USART is fully compatible with the AVR UART regarding:

- Bit locations inside all USART Registers
- Baud Rate Generation
- Transmitter Operation
- Transmit Buffer Functionality
- Receiver Operation

However, the receive buffering has two improvements that will affect the compatibility in some special cases:

- A second buffer register has been added. The two buffer registers operate as a circular FIFO buffer. Therefore the UDR must only be read once for each incoming data! More important is the fact that the error flags (FE and DOR) and the 9th data bit (RXB8) are buffered with the data in the receive buffer. Therefore the status bits must always be read before the UDR Register is read. Otherwise the error status will be lost since the buffer state is lost.
- The receiver Shift Register can now act as a third buffer level. This is done by allowing the received data to remain in the serial Shift Register (see Figure 69) if the buffer registers are full, until a new start bit is detected. The USART is therefore more resistant to Data OverRun (DOR) error conditions.

The following control bits have changed name, but have same functionality and register location:

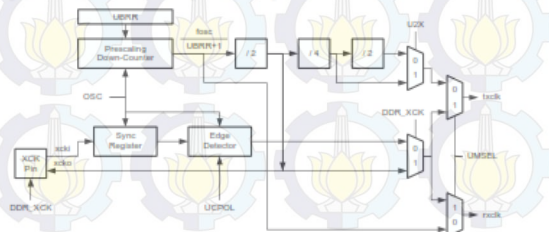
- CHR9 is changed to UCSZ2
- OR is changed to DOR

Clock Generation

The clock generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation: Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous mode. The UMSEL bit in USART Control and Status Register C (UCSRC) selects between asynchronous and synchronous operation. Double Speed (Asynchronous mode only) is controlled by the U2X found in the UCSRA Register. When using Synchronous mode (UMSEL = 1), the Data Direction Register for the XCK pin (DDR_XCK) controls whether the clock source is internal (Master mode) or external (Slave mode). The XCK pin is only active when using Synchronous mode.

Figure 70 shows a block diagram of the clock generation logic.

Figure 70. Clock Generation Logic, Block Diagram



Signal description:

- | | |
|--------------|---|
| txclk | Transmitter clock (Internal Signal). |
| rxclk | Receiver base clock (Internal Signal). |
| input | Input from XCK pin (Internal Signal). Used for synchronous slave operation. |
| xcko | Clock output to XCK pin (Internal Signal). Used for synchronous master operation. |
| fosc | XTAL pin frequency (System Clock). |

Internal Clock Generation – The Baud Rate Generator

Internal clock generation is used for the asynchronous and the synchronous master modes of operation. The description in this section refers to Figure 70.

The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (fosc), is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output ($= \text{fosc}/(\text{UBRR}+1)$). The Transmitter divides the baud rate generator clock output by 2, 8 or 16 depending on mode. The baud rate generator output is used directly by the receiver's clock and data recovery units. However, the recovery units use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSEL, U2X and DDR_XCK bits.

Table 60 contains equations for calculating the baud rate (in bits per second) and for calculating the UBRR value for each mode of operation using an internally generated clock source.

Table 60. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate ¹⁾	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second; bps)

f_{osc} System Oscillator clock frequency

UBRR Contents of the UBRRH and UBRL Registers, (0 - 4095)

Some examples of UBRR values for some system clock frequencies are found in Table 68 (see page 162).

Double Speed Operation (U2X)

The transfer rate can be doubled by setting the U2X bit in UCSRA. Setting this bit only has effect for the asynchronous operation. Set this bit to zero when using synchronous operation.

Setting this bit will reduce the divisor of the baud rate divider from 16 to 8, effectively doubling the transfer rate for asynchronous communication. Note however that the receiver will in this case only use half the number of samples (reduced from 16 to 8) for data sampling and clock recovery, and therefore a more accurate baud rate setting and system clock are required when this mode is used. For the Transmitter, there are no downsides.

External Clock

External clocking is used by the synchronous slave modes of operation. The description in this section refers to Figure 70 for details.

External clock input from the XCK pin is sampled by a synchronization register to minimize the chance of meta-stability. The output from the synchronization register must then pass through an edge detector before it can be used by the Transmitter and receiver. This process introduces a two CPU clock period delay and therefore the maximum external XCK clock frequency is limited by the following equation:

$$f_{XCK} \leq \frac{f_{osc}}{4}$$

Note that f_{osc} depends on the stability of the system clock source. It is therefore recommended to add some margin to avoid possible loss of data due to frequency variations.

Synchronous Clock Operation

When Synchronous mode is used (UMSEL = 1), the XCK pin will be used as either clock input (Slave) or clock output (Master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input (on Rx/D) is sampled at the opposite XCK clock edge of the edge the data output (Tx/D) is changed.

USART Initialization

The USART has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage. For interrupt driven USART operation, the global interrupt flag should be cleared (and interrupts globally disabled) when doing the initialization.

Before doing a re-initialization with changed baud rate or frame format, be sure that there are no ongoing transmissions during the period the registers are changed. The TXC flag can be used to check that the Transmitter has completed all transfers, and the RXC flag can be used to check that there are no unread data in the receive buffer. Note that the TXC flag must be cleared before each transmission (before UDR is written) if it is used for this purpose.

The following simple USART initialization code examples show one assembly and one C function that are equal in functionality. The examples assume asynchronous operation using polling (no interrupts enabled) and a fixed frame format. The baud rate is given as a function parameter. For the assembly code, the baud rate parameter is assumed to be stored in the r17:r16 registers. When the function writes to the UCSRC Register, the URSEL bit (MSB) must be set due to the sharing of I/O location by UBRRH and UCSRC.

Assembly Code Example⁽¹⁾

```
USART_Init:
    ; Set baud rate
    out UBRRH, r17
    out UBRL, r16
    ; Enable receiver and transmitter
    ldi r16, (1<<RXEN)|(1<<TXEN)
    out UCSRB, r16
    ; Set frame format: 8data, 2stop bit
    ldi r16, (1<<URSEL)|(1<<USCR1)|(1<<UCSR0)
    out UCSRC, r16
    ret
```

C Code Example⁽¹⁾

```
void USART_Init( unsigned int baud )
{
    /* Set baud rate */
    UBRRH = (unsigned char)(baud>>8);
    UBRL = (unsigned char)baud;
    /* Enable receiver and transmitter */
    UCSRB = (1<<RXEN)|(1<<TXEN);
    /* Set Frame Format: 8data, 2stop bit */
    UCSRC = (1<<URSEL)|(1<<USCR1)|(1<<UCSR0);
}
```

Note: 1. The example code assumes that the part specific header file is included.

More advanced initialization routines can be made that include frame format as parameters, disable interrupts and so on. However, many applications use a fixed setting of the baud and control registers, and for these types of applications the initialization code can be placed directly in the main routine, or be combined with initialization code for other I/O modules.



Data Transmission – The USART Transmitter

The USART Transmitter is enabled by setting the *Transmit Enable* (TXEN) bit in the UCSRB Register. When the Transmitter is enabled, the normal port operation of the Tx/D pin is overridden by the USART and given the function as the transmitter's serial output. The baud rate, mode of operation and frame format must be set up once before doing any transmissions. If synchronous operation is used, the clock on the XCK pin will be overridden and used as transmission clock.

Sending Frames with 5 to 8 Data Bit

A data transmission is initiated by loading the transmit buffer with the data to be transmitted. The CPU can load the transmit buffer by writing to the UDR I/O location. The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is ready to send a new frame. The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted. When the Shift Register is loaded with new data, it will transfer one complete frame at the rate given by the baud register, U2X bit or by XCK depending on mode of operation.

The following code examples show a simple USART transmit function based on polling of the *Data Register Empty* (UDRE) flag. When using frames with less than eight bits, the most significant bits written to the UDR are ignored. The USART has to be initialized before the function can be used. For the assembly code, the data to be sent is assumed to be stored in Register R16

Assembly Code Example⁽¹⁾

```
USART_Transmit:
; Wait for empty transmit buffer
shlb UCSDR, UDRE
rjmp USART_Transmit
; Put data (r16) into buffer, sends the data
out  UDR, r16
ret
```

C Code Example⁽¹⁾

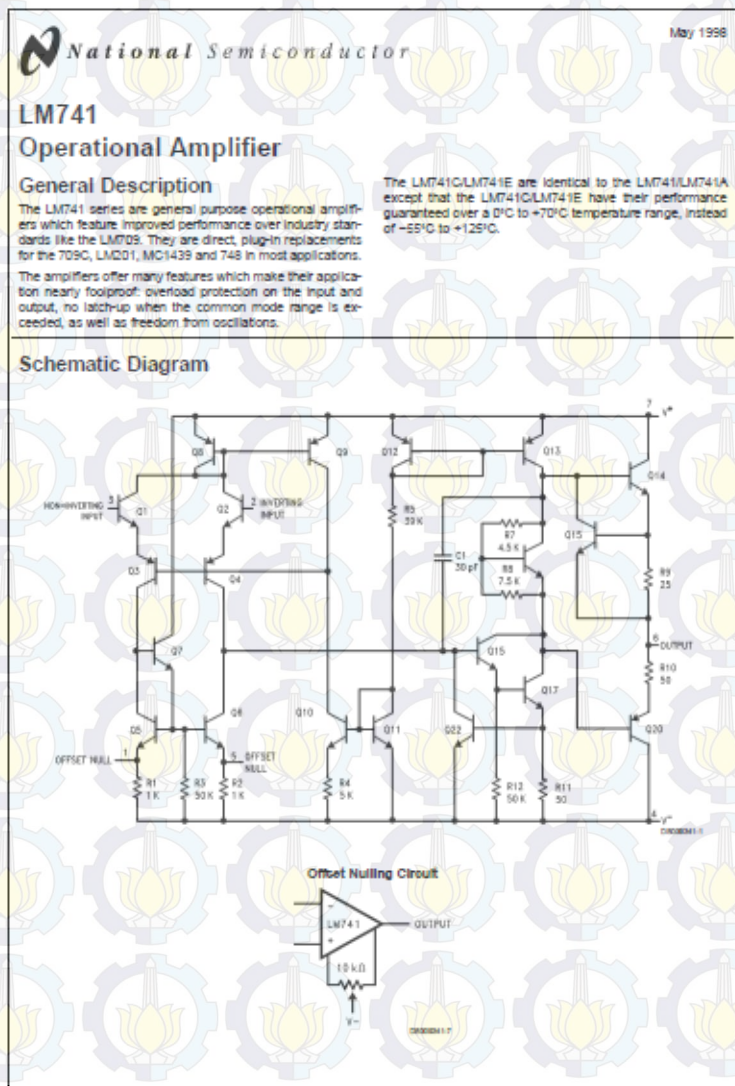
```
void USART_Transmit( unsigned char data )
{
    /* Wait for empty transmit buffer */
    while ( !(UCSDR & (1<<UDRE)) ) {}

    /* Put data into buffer, sends the data */
    UDR = data;
}
```

Note: 1. The example code assumes that the part specific header file is included.

The function simply waits for the transmit buffer to be empty by checking the UDRE flag, before loading it with new data to be transmitted. If the data register empty interrupt is utilized, the interrupt routine writes the data into the buffer.

2. Datasheet LM 741



Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

(Note 5)

	LM741A	LM741E	LM741	LM741C
Supply Voltage	±22V	±22V	±22V	±18V
Power Dissipation (Note 2)	500 mW	500 mW	500 mW	500 mW
Differential Input Voltage	±30V	±30V	±30V	±30V
Input Voltage (Note 3)	±15V	±15V	±15V	±15V
Output Short-Circuit Duration	Continuous	Continuous	Continuous	Continuous
Operating Temperature Range	-55°C to +125°C	0°C to +70°C	-55°C to +125°C	0°C to +70°C
Storage Temperature Range	-65°C to +150°C	-65°C to +150°C	-65°C to +150°C	-65°C to +150°C
Junction Temperature	150°C	100°C	150°C	100°C
Soldering Information				
N-Package (10 seconds)	260°C	260°C	260°C	260°C
J- or H-Package (10 seconds)	300°C	300°C	300°C	300°C
H-Package				
Vapor Phase (60 seconds)	215°C	215°C	215°C	215°C
Infrared (15 seconds)	215°C	215°C	215°C	215°C
See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.				
ESD Tolerance (Note 7)	400V	400V	400V	400V

Electrical Characteristics (Note 4)

Parameter	Conditions	LM741A/LM741E			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	$T_A = 25^\circ\text{C}$										mV
	$R_{th} \leq 10\text{ k}\Omega$				1.0	5.0		2.0	6.0		
	$R_{th} \leq 50\Omega$		0.8	3.0							mV
	$T_{AMIN} \leq T_A \leq T_{AMAX}$			4.0							mV
Average Input Offset Voltage Drift	$R_{th} \leq 50\Omega$						6.0			7.5	mV
	$R_{th} \leq 10\text{ k}\Omega$										$\mu\text{V}/^\circ\text{C}$
Input Offset Current	$T_A = 25^\circ\text{C}$										nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$										
Average Input Offset Current Drift	$T_A = 25^\circ\text{C}$		3.0	30		30	200		20	200	nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$			70		85	500			300	nA
Input Bias Current	$T_A = 25^\circ\text{C}$			0.5							nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$										nA
Input Resistance	$T_A = 25^\circ\text{C}$		30	80		80	500		80	500	nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$			0.210			1.5			0.8	μA
Input Voltage Range	$T_A = 25^\circ\text{C}$		1.0	6.0		0.3	2.0		0.3	2.0	M Ω
	$T_{AMIN} \leq T_A \leq T_{AMAX}$		0.5								M Ω
Input Voltage Range	$T_A = 25^\circ\text{C}$								± 12	± 13	V
	$T_{AMIN} \leq T_A \leq T_{AMAX}$					± 12	± 13				V

Electrical Characteristics (Note 4) (Continued)

Parameter	Conditions	LM741A/LM741E			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain	$T_A = 25^\circ\text{C}$, $R_L \geq 2\text{ k}\Omega$	50			50	200		20	200		V/mV
	$V_B = \pm 20\text{V}$, $V_{CE} = \pm 15\text{V}$										V/mV
	$V_B = \pm 15\text{V}$, $V_{CE} = \pm 10\text{V}$										V/mV
	$T_{AMIN} \leq T_A \leq T_{AMAX}$										V/mV
Output Voltage Swing	$R_L \geq 2\text{ k}\Omega$	32			25			15			V/mV
	$V_B = \pm 20\text{V}$, $V_{CE} = \pm 15\text{V}$										V/mV
	$V_B = \pm 15\text{V}$, $V_{CE} = \pm 10\text{V}$	10									V/mV
	$V_B = \pm 5\text{V}$, $V_{CE} = \pm 2\text{V}$										V/mV
Output Voltage Swing	$V_B = \pm 20\text{V}$	± 15									V
	$R_L \geq 10\text{ k}\Omega$	± 15									V
	$R_L \geq 2\text{ k}\Omega$										V
	$V_B = \pm 15\text{V}$				± 12	± 14		± 12	± 14		V
Output Short Circuit Current	$R_L \geq 10\text{ k}\Omega$				± 10	± 13		± 10	± 13		V
	$R_L \geq 2\text{ k}\Omega$										V
	$V_B = \pm 15\text{V}$	10	25	35				25			mA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$	10		40							mA
Common-Mode Rejection Ratio	$T_{AMIN} \leq T_A \leq T_{AMAX}$										dB
	$R_B \leq 10\text{ k}\Omega$, $V_{CEM} = \pm 12\text{V}$				70	90		70	90		dB
	$R_B \leq 50\Omega$, $V_{CEM} = \pm 12\text{V}$	80	95								dB
	$V_B = \pm 20\text{V}$ to $V_B = \pm 5\text{V}$										dB
Supply Voltage Rejection Ratio	$R_B \leq 50\Omega$	86	96								dB
	$R_B \leq 10\text{ k}\Omega$				77	96		77	96		dB
	$T_{AMIN} \leq T_A \leq T_{AMAX}$										dB
	$V_B = \pm 20\text{V}$ to $V_B = \pm 5\text{V}$										dB
Transient Response	$T_A = 25^\circ\text{C}$, Unity Gain										dB
	Rise Time		0.25	0.8		0.3		0.3			μs
	Settling Time		6.0	20		5		5			μs
	Overshoot										%
Bandwidth (Note 5)	$T_A = 25^\circ\text{C}$	0.437	1.5								MHz
Slew Rate	$T_A = 25^\circ\text{C}$, Unity Gain	0.3	0.7			0.5		0.5			V/ μs
Supply Current	$T_A = 25^\circ\text{C}$				1.7	2.8		1.7	2.8		mA
Power Consumption	$T_A = 25^\circ\text{C}$										mW
	$V_B = \pm 20\text{V}$	80	150								mW
	$V_B = \pm 15\text{V}$				50	85		50	85		mW
	$V_B = \pm 5\text{V}$										mW
LM741A	$T_A = T_{AMIN}$			165							mW
	$T_A = T_{AMAX}$			135							mW
LM741E	$V_B = \pm 20\text{V}$										mW
	$T_A = T_{AMIN}$			150							mW
	$T_A = T_{AMAX}$			150							mW
LM741	$V_B = \pm 20\text{V}$										mW
	$T_A = T_{AMIN}$				60	100					mW
	$T_A = T_{AMAX}$				45	75					mW
	$V_B = \pm 15\text{V}$										mW

Note 1: "Absolute Maximum Ratings" indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits.

3. Datasheet MAX 232

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS
SLL5047L - 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
- ± 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
 MAX232I ... D, DW, OR N PACKAGE
 (TOP VIEW)

C1+	1	16	V _{CC}
V _{SA} -	2	15	GND
C1-	3	14	T1OUT
C2+	4	13	R1IN
C2-	5	12	R1OUT
V _S -	6	11	T1IN
T2OUT	7	10	T2IN
R2IN	8	9	R2OUT

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 ± 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	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
-40°C to 85°C	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
	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	

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

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS

91LS047L - FEBRUARY 1990 - REVISED MARCH 2004

Function Tables

EACH DRIVER

INPUT T1IN	OUTPUT T1OUT
L	H
H	L

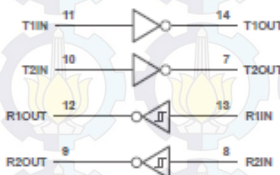
H = high level, L = low level

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**

SL15047L • 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 5 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 (max), θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_{D(max)} = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.

3. The package thermal impedance is calculated in accordance with JE3D 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)		0.8		V
R1IN, R2IN	Receiver input voltage			± 30	V
T_A	Operating free-air temperature				°C
		MAX232	0	70	
		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.

MAX232, MAX231 **DUAL EIA-232 DRIVERS/RECEIVERS**

SLLSD47L - 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 kΩ to GND	5	7		V
V _{OL} Low-level output voltage [‡]	T1OUT, T2OUT R _L = 3 kΩ to GND		-7	-5	V
r _o Output resistance	T1OUT, T2OUT V _{S+} = V _{S-} = 0, V _O = ±2 V	300			Ω
I _{OS} Short-circuit output current	T1OUT, T2OUT V _{CC} = 5.5 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 V, T_A = 25°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 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR Driver slew rate	R _L = 3 kΩ to 7 kΩ, 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		kb/s

NOTE 4: Test conditions are C1-C4 = 1 μF at V_{CC} = 5 V ± 0.5 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 mA	3.5			V
V _{OL} Low-level output voltage [‡]	R1OUT, R2OUT I _{OL} = 3.2 mA			0.4	V
V _{IT+} Receiver positive-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C		1.7	2.4	V
V _{IT-} Receiver negative-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C	0.8	1.2		V
V _{hys} Input hysteresis voltage	R1IN, R2IN V _{CC} = 5 V	0.2	0.5	1	V
η Receiver input resistance	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C	3	5	7	kΩ

[†] All typical values are at V_{CC} = 5 V, T_A = 25°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 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4 and Figure 1)

PARAMETER	TYP	UNIT
t _{PLH(R)} Receiver propagation delay time, low- to high-level output	500	ns
t _{PHL(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 V ± 0.5 V.

4. Datasheet LM7805

FAIRCHILD
SEMICONDUCTOR*

www.fairchildsemi.com

MC78XX/LM78XX/MC78XXA

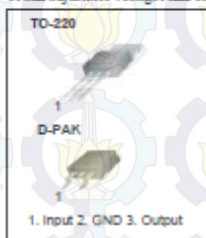
3-Terminal 1A Positive Voltage Regulator

Features

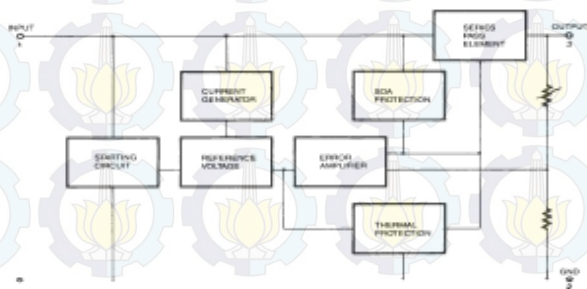
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$) (for $V_O = 24V$)	V_I V_{I1}	35 40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range	T_{OPR}	$0 \sim +125$	$^{\circ}C$
Storage Temperature Range	T_{STG}	$-65 \sim +150$	$^{\circ}C$

Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit, $0^{\circ}C < T_J < 125^{\circ}C$, $I_O = 500mA$, $V_I = 10V$, $C_I = 0.33\mu F$, $C_O = 0.1\mu F$, unless otherwise specified)

Parameter	Symbol	Conditions	MC7805/LM7805			Unit
			Min.	Typ.	Max.	
Output Voltage	V_O	$T_J = +25^{\circ}C$	4.8	5.0	5.2	V
		$5.0mA \leq I_O \leq 1.0A$, $P_O \leq 15W$ $V_I = 7V$ to $20V$	4.75	5.0	5.25	
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}C$	$V_O = 7V$ to $25V$		4.0	mV
			$V_I = 8V$ to $12V$		1.6	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}C$	$I_O = 5.0mA$ to $1.5A$		9	mV
			$I_O = 250mA$ to $750mA$		4	
Quiescent Current	I_Q	$T_J = +25^{\circ}C$	-	5.0	8.0	mA
Quiescent Current Change	ΔI_Q	$I_O = 5mA$ to $1.0A$	-	0.03	0.5	mA
		$V_I = 7V$ to $25V$	-	0.3	1.3	
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5mA$	-	-0.8	-	mV/ $^{\circ}C$
Output Noise Voltage	V_N	$f = 10Hz$ to $100kHz$, $T_A = +25^{\circ}C$	-	42	-	$\mu V/V_O$
Ripple Rejection	RR	$f = 120Hz$ $V_O = 8V$ to $18V$	62	73	-	dB
Dropout Voltage	V_{Drop}	$I_O = 1A$, $T_J = +25^{\circ}C$	-	2	-	V
Output Resistance	r_O	$f = 1kHz$	-	15	-	m Ω
Short Circuit Current	I_{SC}	$V_I = 35V$, $T_A = +25^{\circ}C$	-	230	-	mA
Peak Current	I_{PK}	$T_J = +25^{\circ}C$	-	2.2	-	A

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

5. Datasheet LM7905



November 1994

LM79XX Series 3-Terminal Negative Regulators

General Description

The LM7900 series of 3-terminal regulators is available with fixed output voltages of: $-5V$, $-8V$, $-12V$, and $-15V$. These devices need only one external component—a compensation capacitor at the output. The LM7900 series is packaged in the TO-220 power package and is capable of supplying 1.5A of output current.

These regulators employ internal current limiting, safe area protection and thermal shutdown for protection against virtually all overload conditions.

Low ground pin current of the LM7900 series allows output voltage to be easily boosted above the preset value with a resistor divider. The low quiescent current drain of

these devices with a specified maximum change with line and load ensures good regulation in the voltage boosted mode.

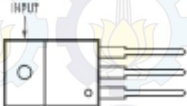
For applications requiring other voltages, see LM317 data sheet.

Features

- Thermal, short circuit and safe area protection
- High ripple rejection
- 1.5A output current
- 4% tolerance on preset output voltage

Connection Diagrams

TO-220 Package

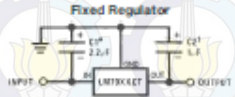


Front View

Order Number LM7905CT, LM7912CT or LM7915CT
See NS Package Number TO3B

Typical Applications

Fixed Regulator



TL4947340-3

*Required if regulator is separated from filter capacitor by more than 8". For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted.

†Required for stability. For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted. Values given may be increased without limit.

For output capacitance in excess of 100 μF , a high current diode from input to output (1N4001, etc.) will protect the regulator from momentary input shorts.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage
($V_{in} = -5V$) $-25V$
($V_{in} = -12V$ and $-15V$) $-35V$

Input-Output Differential

($V_{in} = -5V$) $25V$

($V_{in} = -12V$ and $-15V$) $30V$

Power Dissipation (Note 2) Internally Limited

Operating Junction Temperature Range $0^{\circ}C$ to $+125^{\circ}C$

Storage Temperature Range $-65^{\circ}C$ to $+150^{\circ}C$

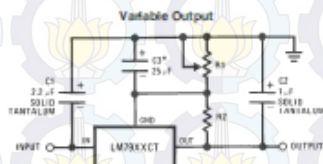
Lead Temperature (Soldering, 10 sec.) $230^{\circ}C$

Electrical Characteristics

Conditions unless otherwise noted: $I_{OUT} = 500$ mA, $C_{IN} = 2.2$ μF , $C_{OUT} = 1$ μF , $0^{\circ}C \leq T_J \leq +125^{\circ}C$, Power Dissipation $\leq 1.5W$.

Part Number			LM7905C			Units
Output Voltage			-5V			
Input Voltage (unless otherwise specified)			-10V			
Symbol	Parameter	Conditions	Min	Typ	Max	
V _O	Output Voltage	T _J = 25°C 5 mA ≤ I _{OUT} ≤ 1 A P ≤ 15W	-4.8	-5.0	-5.2	V
			-4.75		-5.25	V
			(-20 ≤ V _{IN} ≤ -7)			V
ΔV _O	Line Regulation	T _J = 25°C (Note 3)		8	50	mV
			(-25 ≤ V _{IN} ≤ -7)			V
				2	15	mV
(-12 ≤ V _{IN} ≤ -8)					V	
ΔV _O	Load Regulation	T _J = 25°C (Note 3) 5 mA ≤ I _{OUT} ≤ 1.5A 250 mA ≤ I _{OUT} ≤ 750 mA		15	100	mV
				5	50	mV
I _O	Quiescent Current	T _J = 25°C		1	2	mA
ΔI _O	Quiescent Current Change	With Load			0.5	mA
		With Load, 5 mA ≤ I _{OUT} ≤ 1A			0.5	V
V _n	Output Noise Voltage	T _A = 25°C, 10 Hz ≤ f ≤ 100 Hz		125		μV
	Ripple Rejection	f = 120 Hz	54	66		dB
			(-18 ≤ V _{IN} ≤ -8)			V
	Dropout Voltage	T _J = 25°C, I _{OUT} = 1A		1.1		V
I _{O(MAX)}	Peak Output Current	T _J = 25°C		2.2		A
	Average Temperature Coefficient of Output Voltage	I _{OUT} = 5 mA, 0°C ≤ T _J ≤ 100°C		0.4		mV/°C

Typical Applications (Continued)



*Improve transient response and ripple rejection. Do not increase beyond 50 μF .

$$V_{OUT} = V_{REF} \left(\frac{R1 + R2}{R2} \right)$$

Select $R2$ as follows:

LM7905CT 300Ω

LM7910CT 750Ω

LM7915CT $1k$

TL431/540-2

6. Tarif Tenaga Listrik Juni 2016

 PT PLN (PERSERO) Jalan Trunojoyo Blok M 1/135 Kebayoran Baru - Jakarta 12160 Telp. : (021) 7261875, 7261122, 7262234 (021) 7251234, 7250550 Kotak Pos : 4322/KBB Faximile : (021) 7221330 Alamat Kawat : PLNPT					
PENETAPAN PENYESUAIAN TARIF TENAGA LISTRIK (TARIFF ADJUSTMENT) BULAN JUNI 2016					
NO.	GOL. TARIF	BATAS DAYA	BIAYA BEBAN (Rp/kVA/bulan)	REGULER BIAYA PEMAKAIAN (Rp/kWh) DAN BIAYA kWh (Rp/kVAh)	PRA BAYAR (Rp/kWh)
1.	R-1/TR	1.300 VA	*)	1.364,86	1.364,86
2.	R-1/TR	2.200 VA	*)	1.364,86	1.364,86
3.	R-2/TR	3.500 VA s.d. 5.500 VA	*)	1.364,86	1.364,86
4.	R-3/TR	6.600 VA ke atas	*)	1.364,86	1.364,86
5.	B-2/TR	6.600 VA s.d. 200 kVA	*)	1.364,86	1.364,86
6.	B-3/TM	di atas 200 kVA	**)	Blok WBSP = $K \times 975,49$ Blok LWBP = $975,49$ kVAh = 1.049,85 ****)	-
7.	I-3/TM	di atas 200 kVA	**)	Blok WBSP = $K \times 975,49$ Blok LWBP = $975,49$ kVAh = 1.049,85 ****)	-
8.	I-4/TT	30.000 kVA ke atas	***)	Blok WBSP dan Blok LWBP = $939,85$ kVAh = $939,85$ ****)	-
9.	P-1/TR	6.600 VA s.d. 200 kVA	*)	1.364,86	1.364,86
10.	P-2/TM	di atas 200 kVA	**)	Blok WBSP = $K \times 975,49$ Blok LWBP = $975,49$ kVAh = 1.049,85 ****)	-
11.	P-3/TR	di atas 200 kVA	*)	1.364,86	1.364,86
12.	L/TR, TM, TT			1.542,63	-
Catatan : *) Diterapkan Rekening Minimum (RM). RM1 = 40 (Jam Nyala) x Daya tersambung (kVA) x Biaya Pemakaian. **) Diterapkan Rekening Minimum (RM). RM2 = 40 (Jam Nyala) x Daya tersambung (kVA) x Biaya Pemakaian LWBP. Jam nyala : kWh per bulan dibagi dengan kVA tersambung. ***) Diterapkan Rekening Minimum (RM). RM3 = 40 (Jam Nyala) x Daya tersambung (kVA) x Biaya Pemakaian WBP dan LWBP. Jam nyala : kWh per bulan dibagi dengan kVA tersambung. ****) Biaya kelebihan pemakaian daya reaktif (kVAh) dikenakan dalam hal faktor daya rata-rata setiap bulan kurang dari 0,85 (delapan puluh lima per seratus). K : Faktor perbandingan antara harga WBP dan LWBP sesuai dengan karakteristik beban sistem kelistrikan setempat ($1,4 \leq K \leq 2$), ditetapkan oleh Direksi Perusahaan Perseroan (Persero) PT Perusahaan Listrik Negara. WBP : Waktu Beban Puncak. LWBP : Luar Waktu Beban Puncak.					

DAFTAR RIWAYAT HIDUP



Nama : Betty Nur Qamarina
TTL : Kediri, 28 April 1995
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Perum Wilis Indah II Blok
H-8 No.34, Kediri
Telp/HP : 0856492103482
E-mail : ungsyaam@gmail.com

RIWAYAT PENDIDIKAN

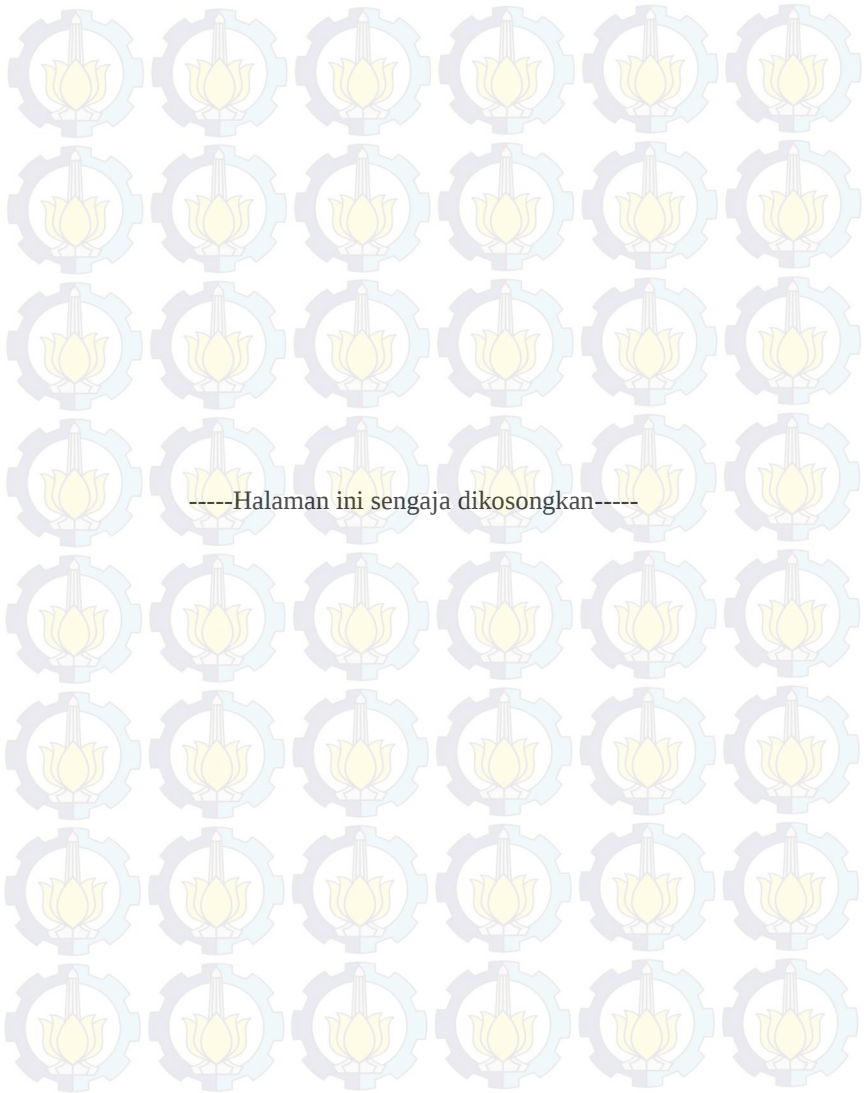
1. 2001 – 2007 : SD Negeri Sukorame 3 Kediri
2. 2007 – 2010 : SMP Negeri 4 Kediri
3. 2010 – 2013 : SMA Negeri 1 Kediri
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PG. Pesantren Kediri
2. Kerja Praktek di PT. PLN (Persero) Area Surabaya Barat

PENGALAMAN ORGANISASI

1. Staff Departemen Komunikasi dan Informasi Periode 2014/2015 HIMAD3TEKTRO, FTI - ITS
2. Kabiromo Medfo Departemen Komunikasi dan Informasi Periode 2015/2016 HIMAD3TEKTRO, FTI - ITS



DAFTAR RIWAYAT HIDUP



Nama : Medif Zepta Yogassasena
TTL : Lamongan, 16 September 1996
Jenis Kelamin : Laki - Laki
Agama : Islam
Alamat : Bandut 02/01
Bogobabadan
Karangbinangun
Lamongan
Telp/HP : 081233147776
E-mail : medifzepta508@gmail.com

RIWAYAT PENDIDIKAN

1. 2001 – 2007 : MI Ihyaul Ulum Gresik
2. 2007 – 2010 : SMP Negeri 2 Karangbinangun
3. 2010 – 2013 : SMA Unggulan BPPT Al-Fattah
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Miwon Indonesia
2. Kerja Praktek di PT. PJB UP Gresik

PENGALAMAN ORGANISASI

