



**TUGAS AKHIR - TE095561**

**PROTOTIPE PENGIDENTIFIKASI JENIS PERALATAN  
ELEKTRONIK RUMAH TANGGA MENGGUNAKAN  
SPEKTRUM HARMONISA ARUS**

Ahmad Naufa Syauqi  
NRP 2212 038 023  
Abdul Hadi  
NRP 2212 038 027

Dosen Pembimbing  
Dr. Ardyono Priyadi, ST., M.Eng

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



*FINAL PROJECT - TE145561*

***PROTOTYPE FOR IDENTIFY THE TYPE OF  
HOUSEHOLD ELECTRONIC EQUIPMENT USING  
CURRENT HARMONIC SPECTRUM***

Ahmad Naufa Syauqi  
NRP 2212 038 023  
Abdul Hadi  
NRP 2212 038 027

*Conseullor*  
Dr. Ardyono Priyadi, ST., M.Eng

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

**PROTOTYPE PENGIDENTIFIKASI JENIS PERALATAN  
ELEKTRONIK RUMAH TANGGA MENGGUNAKAN  
SPEKTRUM HARMONISA ARUS**

**TUGAS AKHIR**

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

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

**Menyetujui :  
Dosen Pembimbing,**



**Dr. Ardyono Priyadi, ST., M.Eng**  
**NIP. 19730927 199803 1 004**

**SURABAYA  
JULI, 2015**

# PROTOTYPE PENGIDENTIFIKASI JENIS PERALATAN ELEKTRONIK RUMAH TANGGA MENGGUNAKAN SPEKTRUM HARMONISA ARUS

**Nama Mahasiswa** : 1. Ahmad Naufa Syauchi  
2. Abdul Hadi  
**NRP** : 1. 2212 038 023  
2. 2212 038 027  
**Dosen Pembimbing** : Dr. Ardyono Priyadi, ST., M.Eng.  
**NIP** : 19730927 199803 1 004

## ABSTRAK

Penggunaan beban – beban non linier seperti mesin-mesin sinkron, lampu hemat energi, las dan lain-lain dapat menyebabkan gelombang arus dan tegangan pada sistem tenaga listrik terdistorsi atau yang sering disebut harmonisa yang mengakibatkan keamanan dan kualitas dari sistem tenaga listrik terganggu. Harmonisa merupakan distorsi periodik dari gelombang sinus tegangan, arus, atau daya dengan bentuk gelombang yang frekuensinya merupakan kelipatan diluar bilangan satu terhadap frekuensi fundamental (frekuensi 50 Hz/60 Hz). Akibat lain yang ditimbulkan harmonisa adalah rusaknya peralatan listrik, penurunan daya trafo & bertambahnya *losses*, dan terjadinya kesalahan ukur pada kWh meter.

Oleh karena itu, perlu dibuat alat untuk memonitoring harmonisa dan kemudian diidentifikasi peralatan elektronik yang menimbulkan harmonisa. Pada tugas akhir ini dibuat alat pendeteksi harmonisa arus menggunakan metode *Discrete Fourier Transform* (DFT) berbasis mikrokontroler. Mikro akan mengirimkan data serial ke komputer dan akan diolah oleh aplikasi *borland delphi* untuk diketahui harmonisa yang ada pada peralatan elektronik rumah tangga. Kemudian identifikasi hasil THDi di *Visual Basic*.

Setelah dilakukan *monitoring* dan identifikasi dapat diketahui peralatan elektronik yang ada di rumah tangga dapat menimbulkan harmonisa atau tidak. Hasil pengukuran yang didapat antara prototipe dan alat ukur memiliki selisih *%error* yang kecil sehingga prototipe layak untuk digunakan. Prototipe mampu mendeteksi keberadaan harmonisa arus hingga orde 15 atau pada frekuensi 750Hz.

Kata kunci : Harmonisa arus, DFT, *delphi 7*, Identifikasi

*Halaman ini sengaja dikosongkan*

## **PROTOTYPE FOR IDENTIFY THE TYPE OF HOUSEHOLD ELECTRONIC EQUIPMENT USING CURRENT HARMONIC**

**Name of Student** : 1. Ahmad Naufa Syaui  
2. Abdul Hadi  
**NRP** : 1. 2212 038 023  
2. 2212 038 027  
**Supervisor** : Dr. Ardyono Priyadi, ST., M.Eng.  
**NIP** : 19730927 199803 1 004

### **ABSTRACT**

*The use of non linier loads such as synchronous machines, energy saving lamps, welds, and so on can create a surge current and voltage of the power system distorted or often called harmonics resulting in safety and quality of electric power systems disrupted. Harmonic is periodic distortion of the sine wave voltage, current, or power with waveform, which frequency is multiple of the number higher than one, on the outside of the fundamental frequency (frequency 50 Hz/60Hz). The other effect of the harmonic can damage electrical equipment, decreasing power of transformer and increasing losses, and error measurement Kwh meter.*

*There for, it is necessary to make a tool for harmonic monitoring and then to identify which electronic equipment caused harmonic. This research make tool that can identify current harmonic using Discrete Fourier Transform (DFT) based on microcontroller. Micro will send serial data to the computer and will be processed by Borland Delphi software to know the amount of harmonics on household electronic equipment. Moreover, identification of the result of THDi will be processed using Visual Basic.*

*After monitoring and identification we can know the electronic equipment in the household can create to harmonics or not. Measurement results obtained between prototype and measuring devices have a small %error so that the prototype feasible for use. The prototype is able to detect the presence of harmonic currents up to order 15 or at a frequency of 750Hz.*

**Keywords** : Current harmonic, DFT, delphi 7, Identifications

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji Syukur kehadirat Allah Subhanahu wa ta'ala atas segala rahmat dan hidayah serta segala nikmat yang telah diberikan-Nya, sehingga tersusunlah Tugas Akhir berjudul :

**“PROTOTIPE PENGIDENTIFIKASI JENIS PERALATAN ELEKTRONIK RUMAH TANGGA MENGGUNAKAN SPEKTRUM HARMONISA ARUS”**

Tugas Akhir ini merupakan salah satu kurikulum sebagai persyaratan yang harus dilengkapi guna menyelesaikan studi di jurusan Diploma-3 (D3) Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.

Penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari kesempurnaan,

Banyak dorongan dan bantuan yang penulis dapatkan selama penyusunan Laporan Tugas Akhir, oleh karena itu penulis banyak terima kasih kepada :

1. Kedua orang tua dan seluruh keluarga atas limpahan doa, dukungan moral dan materil yang telah diberikan kepada penulis
2. Dr. Ardyono Priyadi, ST., M.Eng sebagai dosen pembimbing, yang telah berkenan meluangkan waktu, tenaga, pikiran serta memberikan pengarah dan bimbingan dari awal hingga terselesainya Tugas Akhir
3. Rekan-rekan mahasiswa ITS yang telah banyak membantu dalam penyelesaian Tugas Akhir.

Semoga laporan ini dapat bermanfaat bagi semua terutama bagi penulis. Penulis menyadari bahwa dalam penyusunan Buku Tugas Akhir ini masih terdapat banyak kekurangan. Oleh karena itu saran dan kritik yang membangun sangat penulis harapkan.

Surabaya, Juli 2015

Penulis



*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

<b>JUDUL</b> .....	i
<b>PENGESAHAN</b> .....	iii
<b>ABSTRAK</b> .....	v
<b>ABSTRACT</b> .....	vii
<b>KATA PENGANTAR</b> .....	ix
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xiii
<b>DAFTAR TABEL</b> .....	xvii
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Batasan Masalah .....	2
1.4 Maksud dan Tujuan .....	3
1.5 Sistematika Laporan .....	3
1.6 Relevansi .....	4
<b>BAB II TEORI PENUNJANG</b> .....	5
2.1 Harmonisa .....	5
2.1.1 Efek Keberadaan Harmonik Arus .....	6
2.2 Analisa Harmonisa.....	7
2.2.1 Pengenalan Transformasi <i>Fourier</i> .....	7
2.2.2 Pengembangan Algoritma DFT.....	8
2.3 Indikator dan Standar Distorsi Harmonik Arus .....	9
2.4 Mikrokontroler AVR.....	9
2.4.1 Mikrokontroler At-Mega 16.....	10
2.5 Sensor Arus ACS 712 05 .....	11
2.6 Komunikasi <i>Serial</i> RS-232 .....	12
2.7 Pemograman Grafis Menggunakan <i>Delphi</i> .....	14
2.7.1 Peralatan pada <i>Delphi</i> .....	14
2.7.2 <i>Code Edition</i> .....	15
2.7.3 <i>Form</i> .....	15
2.7.4 <i>Component Palette</i> .....	16
2.7.5 <i>Object Inspector</i> .....	16
2.7.6 <i>Watch List</i> .....	16
2.7.7 Tombol – tombol <i>SortCut</i> .....	16

<b>BAB III PERANCANGAN ALAT .....</b>	<b>19</b>
3.1 Perancangan dan Pembuatan <i>Hardware</i> .....	20
3.1.1 Mikrokontroler.....	20
3.1.2 <i>Downloader</i> .....	21
3.1.3 Komunikasi <i>Serial</i> RS-232 ke PC.....	22
3.1.4 <i>Power Supply</i> .....	24
3.2 Perancangan Pembuatan Software .....	25
3.2.1 Perancangan Program dengan <i>Code Vision AVR</i> .....	25
3.2.2 Pembuatan Software Analisis dan Monitoring .....	28
<b>BAB IV PENGUJIAN DAN ANALISA DATA .....</b>	<b>33</b>
4.1 Pengujian Tegangan <i>Output Power Supply</i> .....	33
4.2 Pengujian Rangkaian Mikrokontroler.....	34
4.3 Pengujian Sensor Arus .....	35
4.4 Pengujian Komunikasi <i>Serial</i> ke PC.....	38
4.5 Pengujian Alat keseluruhan ke Delphi .....	39
4.5.1 Pengujian Menggunakan Lampu Pijar sebagai Beban .....	42
4.5.2 Pengujian Hasil DFT.....	43
4.5.3 Penguian Level THDi.....	44
4.5.4 Identifikasi Peralatan Elektronik Rumah Tangga Menggunakan Matlab .....	46
<b>BAB V PENUTUP .....</b>	<b>49</b>
5.1 Kesimpulan .....	49
5.2 Saran .....	49
<b>DAFTAR PUSTAKA .....</b>	<b>51</b>
<b>LAMPIRAN 1 LISTING PROGRAM.....</b>	<b>A-1</b>
<b>LAMPIRAN 2 LISTING DATASHEET .....</b>	<b>B-1</b>
<b>DAFTAR RIWAYAT HIDUP .....</b>	<b>C-1</b>

## DAFTAR TABEL

Tabel 2.1	Alokasi pin <i>Interface</i> .....	12
Tabel 2.2	Keterangan Pin DB-9 .....	13
Tabel 2.3	Tombol <i>Shortcut</i> .....	17
Tabel 3.1	Karakteristik <i>Serial RS-232</i> .....	23
Tabel 4.1	Hasil pengujian <i>power supply</i> .....	33
Tabel 4.2	Pengujian mikrokontroler.....	34
Tabel 4.3	Data Pengujian Pertama Sensor Arus .....	35
Tabel 4.4	Data Pengujian Kedua Sensor Arus .....	37
Tabel 4.5	Data Tampilan Arus pada <i>Delphi</i> .....	42
Tabel 4.6	Data perbandingan THDi Delphi dengan Fluke.....	45
Tabel 4.7	Data hasil uji peralatan rumah tangga .....	45

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1	Gelombang Fundamental.....	5
Gambar 2.2	Bentuk gelombang dasar, harmonik, dan gelombang terdistorsinya .....	6
Gambar 2.3	Sensor Arus ACS 712 .....	11
Gambar 2.4	Konfigurasi Pin DB-9 .....	13
Gambar 2.5	Tampilan pada <i>Software Delphi</i> .....	15
Gambar 3.1	Diagram <i>blok</i> modul pendeteksian.....	19
Gambar 3.2	Diagram Fungsional <i>Hardware</i> .....	20
Gambar 3.3	Skematik rangkaian mikrokontroler.....	21
Gambar 3.4	Skematik rangkaian <i>downloader</i> .....	22
Gambar 3.5	Kabel <i>serial</i> RS-232.....	23
Gambar 3.6	Skematik rangkaian komunikasi <i>serial</i> .....	24
Gambar 3.7	Rangkaian power supply .....	25
Gambar 3.8	<i>Flowchart code vision</i> AVR.....	26
Gambar 3.9	Pengaturan <i>chip</i> dan <i>clock</i> .....	27
Gambar 3.10	Pengaturan jumlah <i>bit</i> ADC serta <i>sampling rate</i> dengan kristal <i>clock</i> 12 MHz.....	27
Gambar 3.11	Pengaturan Komunikasi <i>Serial</i> .....	28
Gambar 3.12	<i>Listing</i> program DFT .....	29
Gambar 3.13	<i>Listing</i> program mencari nilai THDi.....	30
Gambar 3.14	Flowchart program <i>delphi</i> .....	31
Gambar 3.15	Tampilan <i>monitoring</i> harmonisa.....	32
Gambar 3.16	Tampilan Identifikasi .....	32
Gambar 4.1	Pengujian mikrokontroler.....	34
Gambar 4.2	Grafik hasil arus dari sensor arus ACS 712.....	36
Gambar 4.3	Grafik hasil tegangan <i>Output</i> ACS 712.....	36
Gambar 4.4	Hasil grafik uji coba arus di lampu LHE.....	37
Gambar 4.5	Hasil grafik uji coba tegangan di lampu LHE.....	38
Gambar 4.6	Hasil pengiriman data ke PC dengan serial RS-232	38
Gambar 4.7	<i>Wiring</i> prototipe dengan PC .....	39
Gambar 4.8	Setting koneksi delphi dengan <i>hardware</i> .....	40
Gambar 4.9	Delphi sudah terhubung dengan <i>hardarware</i> .....	40
Gambar 4.10	bentuk gelombang arus pada beban linier diprototipe dari <i>sampel</i> 1 – 150.....	41
Gambar 4.11	Hasil data <i>sampling</i> .....	42
Gambar 4.12	Nilai frekuensi dari frekuensi dasar dari hasil proses pengukuran beban linier.....	43
Gambar 4.13	Harmonisa keluar pada kelipatan 50 Hz.....	44

Gambar 4.14 Hasil Identifikasi Menggunakan *Visual Basic* ..... 46

# BAB I

## PENDAHULUAN

Pada BAB I ini akan dijelaskan mengenai latar belakang pengambilan judul Tugas Akhir, permasalahan yang saat ini terjadi, dan alasan – alasan kenapa judul ini diangkat sebagai judul Tugas Akhir.

### 1.1 Latar Belakang

Kenaikan tarif dasar listrik menyebabkan banyak masyarakat yang menggunakan peralatan elektronik yang hemat energi. Peralatan – peralatan hemat energi tersebut merupakan beban non linier. Beban non linier adalah bentuk gelombang keluarannya tidak sebanding dengan tegangan dalam setiap setengan siklus sehingga bentuk gelombang arus maupun tegangan keluarannya tidak sama dengan gelombang masukannya (mengalami distorsi). Penggunaan beban – beban non linier seperti yang terdapat pada *inverter*, *rectifier*, lampu *fluorescent* dengan elektronik *ballast*, komputer, mesin *fotocopy*, mesin fax, televisi, UPS, *Variable Speed Drive* (VSD) untuk motor asinkron atau motor DC menyebabkan gelombang arus dan tegangan pada sistem tenaga listrik terdistorsi atau yang sering disebut harmonisa yang mengakibatkan keamanan dan kualitas dari sistem tenaga listrik terganggu.

Harmonisa merupakan distorsi periodik dari gelombang sinus tegangan, arus, atau daya dengan bentuk gelombang yang frekuensinya merupakan kelipatan diluar bilangan satu terhadap frekuensi fundamental (frekuensi 50 Hz/60 Hz). Nilai frekuensi dari gelombang harmonisa yang terbentuk merupakan hasil kali antara frekuensi fundamental dengan bilangan harmonisanya ( $f$ ,  $2f$ ,  $3f$ , dst). Bentuk gelombang yang terdistorsi merupakan penjumlahan dari gelombang fundamental dan gelombang harmonisa ( $h_1$ ,  $h_2$ , dan seterusnya) pada frekuensi kelipatannya. Semakin banyak gelombang harmonisa yang diikutsertakan pada gelombang fundamentalnya, maka gelombang akan semakin mendekati gelombang persegi atau gelombang akan berbentuk non sinusoidal.

Banyak kerugian yang diakibatkan oleh timbulnya harmonisa salah satunya adalah pada sistem distribusi, aliran harmonisa dapat menurunkan kualitas daya (*Power Quality*) dan menyebabkan timbulnya beberapa masalah seperti terjadinya *overload* sistem akibat kenaikan nilai arus *r.m.s*, *overload* konduktor netral, peningkatan temperatur pada konduktor yang bisa mengakibatkan terbakarnya konduktor, terjadi salah ukur pada *KWH*-meter dan menurunnya umur pakai dan keandalan



peralatan tenaga listrik yang digunakan. Alat yang digunakan untuk melakukan pengukuran harmonisa adalah *power quality analyzer* tetapi mempunyai kelemahan yaitu *power quality analyzer* memiliki harga yang mahal, tidak dapat dipasang secara permanen untuk melakukan pemantauan dan terbatas penggunaannya pada satu produsen yang sama (satu jenis merk) sehingga sulit terintegrasi dengan sistem SCADA.

Oleh karena itu, pada buku ini akan membuat prototipe pendeteksi harmonisa dengan menggunakan ACS 712 untuk mendapatkan data analog dari gelombang arus pada sistem tenaga listrik dan metode analisis *Discrete Fourier Transform* (DFT) untuk mengubah gelombang arus dalam domain waktu menjadi domain frekuensi. Prototipe menggunakan mikrokontroler sebagai *interface* antara ACS 712 dengan *Personal Computer* yang bertujuan untuk mengolah data analog menjadi data digital untuk dianalisis di *Personal Computer*. Borland Delphi sebagai *software* untuk menampilkan gelombang *output* ADC, gelombang arus, nilai arus r.m.s, spektrum arus dalam domain frekuensi dan nilai THDi.

## 1.2 Permasalahan

Saat ini penggunaan beban *non* linier, lampu hemat energi dan peralatan elektronik lainnya dapat mengakibatkan terjadinya harmonisa arus pada sistem jaringan distribusi. Harmonisa arus ini akan memiliki dampak negatif bagi pelanggan listrik maupun penjualnya sendiri. Untuk itu, perlu dilakukan identifikasi terhadap peralatan rumah tangga yang berpotensi menghasilkan harmonisa arus, agar tidak menimbulkan kerusakan pada trafo distribusi serta peralatan elektronik yang lain. Alat yang sudah ada seperti *power quality analyzer* untuk mengukur harmonisa mempunyai harga yang mahal sehingga jarang yang memiliki.

## 1.3 Batasan Masalah

Dalam perancangan ini perlu diberikan beberapa batasan permasalahan dengan tujuan agar pembahasan tidak meluas dan menyimpang dari tujuan awal pembuatan alat. Adapun batasan permasalahan dari alat yang dirancang pada Tugas Akhir ini yaitu :

1. *Monitoring* adanya harmonisa arus didalam peralatan elektronik.
2. *Monitoring* harmonisa dilakukan dengan mendeteksi arus pembebanan satu fasa.
3. *Monitoring* dilakukan secara *offline*, sehingga data disimpan terlebih dahulu dan tidak *real time*.

4. *Monitoring* dilakukan secara terjadwal pada waktu yang telah ditentukan.
5. Sensor arus yang digunakan menggunakan *DT-Sense ACS 712* dengan spesifikasi arus maksimal sebesar 5 ampere.
6. Untuk pengambilan sampel data arus, diambil sebanyak 2.500 data.
7. Proses *Analog to Digital Converter (ADC)* menggunakan mikrokontroler.
8. Bahasa pemrograman yang digunakan pada mikrokontroler menggunakan bahasa C pada *Code Vision AVR*.
9. Komunikasi yang digunakan menggunakan komunikasi Serial.
10. *Interfacing* menggunakan software borland delphi.
11. Tampilan pada komputer antara lain yaitu muncul bentuk gelombang arus yang terjadi dan bentuk harmonisa dari tampilan arus.

#### **1.4 Maksud dan Tujuan**

Tujuan dari Tugas Akhir ini adalah:

1. Membuat sebuah *prototype* yang berfungsi untuk mendeteksi gangguan pada jaringan distribusi akibat dari harmonisa arus yang terlalu tinggi.
2. Memonitor peralatan elektronik yang ada di rumah tangga untuk mengetahui peralatan elektronik tersebut dapat menimbulkan harmonisa atau tidak.
3. Dapat mengetahui historis harmonisa yang ditimbulkan oleh suatu peralatan elektronik yang telah di *monitoring*.
4. Dapat mencegah kerusakan secara dini akibat yang ditimbulkan dari harmonisa.
5. Identifikasi harmonisa yang ditimbulkan pada peralatan elektronik rumah tangga.

#### **1.5 Sistematika Laporan**

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

BAB I : PENDAHULUAN

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

BAB II : TEORI PENUNJANG

Berisi teori penunjang yang mendukung dalam perencanaan dan pembuatan alat.

#### BAB III : PERANCANGAN ALAT

Di bab ini dijelaskan tentang perancangan perangkat keras yang meliputi rangkaian-rangkaian elektronik dan *prototype*, perancangan perangkat lunak yang meliputi program yang akan digunakan untuk mengolah data dari alat tersebut.

#### BAB IV : PENGUJIAN DAN ANALISA ALAT

Di bab ini dibahas tentang pengukuran, pengujian, dan analisa terhadap prinsip kerja dan proses dari alat yang dibuat.

#### BAB V : PENUTUP

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

### **1.6 Relevansi**

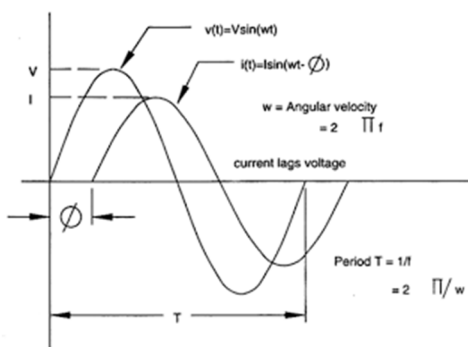
Diharapkan kedepannya dari alat ini dapat dikembangkan untuk dapat membantu PT.PLN (Persero) dalam memonitor masalah gangguan harmonis arus yang terjadi pada jaringan distribusi dan juga membantu dalam pemeliharaan kestabilan pada KWh meter pelanggan. Disamping itu dapat juga membantu para pegawai PT.PLN dalam melaksanakan Sistem *Monitoring* Trafo (SIMONTRA) yang telah ada sehingga dapat mengetahui keadaan dari trafo tersebut.

## BAB II TEORI PENUNJANG

Pada bab ini membahas tentang teori dasar dan teori penunjang dari peralatan-peralatan yang digunakan dalam pembuatan alat pengidentifikasi harmonisa pada peralatan elektronik rumah tangga.

### 2.1 Harmonisa [1]

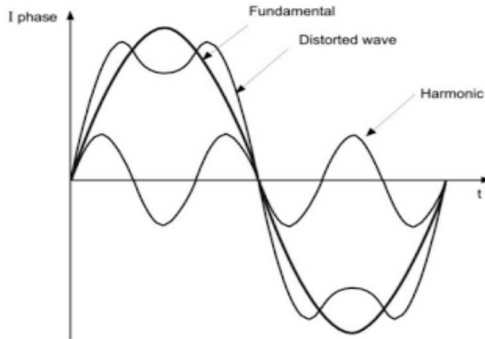
Dalam sistem tenaga listrik yang ideal, bentuk gelombang tegangan yang disalurkan dan bentuk gelombang arus yang dihasilkan adalah gelombang sinus murni, seperti yang ditunjukkan pada gambar 2.1.



Gambar 2.1 Gelombang Fundamental

Akan tetapi dalam kenyataannya bentuk gelombang arus dan tegangan bukanlah gelombang sinus murni. Hal ini dikarenakan adanya distorsi pada gelombang – gelombang tersebut atau yang biasa kita kenal dengan sebutan harmonik. Harmonik adalah gangguan yang terjadi pada sistem distribusi tenaga listrik akibat terjadinya distorsi gelombang arus dan tegangan. Pada dasarnya harmonik adalah gejala pembentukan gelombang – gelombang dengan frekuensi berbeda yang merupakan perkalian bilangan bulat dengan frekuensi dasarnya. Hal ini disebut frekuensi harmonik yang timbul pada bentuk gelombang aslinya sedangkan bilangan bulat pengali frekuensi dasar disebut angka urutan harmonik.

Gelombang – gelombang tersebut kemudian menumpang pada gelombang murni atau aslinya sehingga terbentuk gelombang cacat yang merupakan hasil penjumlahan antara gelombang murni sesaat dengan gelombang harmoniknya, seperti tampak pada gambar 2.2.



**Gambar 2.2** Bentuk gelombang dasar, harmonik, dan gelombang distorsinya

Harmonik ini muncul akibat adanya beban – beban non linier yang terhubung ke sistem distribusi. Beban – beban ini umumnya adalah peralatan elektronik yang didalamnya banyak terdapat komponen – komponen semi konduktor. Komponen ini dalam proses kerjanya berlaku sebagai saklar yang bekerja pada setiap siklus gelombang tegangan.

### 2.1.1 Efek Keberadaan Harmonik Arus

Keberadaan harmonik arus dalam suatu sistem tenaga listrik memberikan efek secara langsung maupun tidak langsung terhadap kualitas dan keandalan sistem tersebut. Dalam sistem distribusi tenaga listrik aliran harmonik menyebabkan terjadinya penurunan kualitas daya (*power quality*) dan beberapa masalah seperti meningkatnya arus rms sehingga terjadi *overload* pada kondisi pembebanan yang seharusnya masih mampu disuplai sistem.

Pengaruh keberadaan harmonik juga sangat tampak pada peralatan – peralatan sistem tenaga listrik seperti generator, transformator, dan motor. Pada generator dampak yang ditimbulkan oleh harmonik adalah peningkatan rugi – rugi tambahan. Rugi – rugi tambahan ini disebabkan oleh rugi – rugi arus urutan nol dan rugi – rugi arus urutan negatif yang muncul akibat harmonik. Untuk pada transformator sendiri, dampak yang ditimbulkan oleh harmonik adalah meningkatnya suhu yang disebabkan oleh arus yang besar dari arus urutan nol yang bersikulasi pada belitan delta. Dengan meningkatnya suhu pada trafo, dapat menyebabkan bertambahnya rugi rugi daya sehingga mengurangi kapasitas pembebanan trafo. Selain itu, dapat mengurangi kemampuan arus maksimum dan umur trafo. Sedangkan

pada motor, harmonik menyebabkan peningkatan rugi – rugi pada belitan stator, rangkaian rotor, serta laminasi stator dan rotor sehingga efisiensi mesin menurun. Yang mana dampak – dampak tersebut berujung pada menyusutnya umur pemakaian peralatan – peralatan tersebut.

Dari segi ekonomi, pengaruh keberadaan harmonik arus juga perlu mendapatkan perhatian serius. Hal ini dikarenakan beberapa alasan diantaranya, pertama penurunan umur pakai peralatan – peralatan sistem tenaga listrik akibat adanya harmonik menyebabkan harus dilakukannya pergantian lebih dini daripada seharusnya. Hal ini sangat merugikan bagi produsen listrik karena disamping mengeluarkan biaya guna mengganti peralatan tersebut, juga mengalami kesulitan dalam perhitungan biaya dan berpotensi menimbulkan kerugian. Kedua, terjadinya *overload* akibat meningkatnya arus rms yang disebabkan oleh harmonik harus diimbangi dengan penambahan kapasitas pembangkit dan saluran transmisi maupun distribusi guna menjaga keandalan dari sistem. Padahal kebutuhan daya yang menyebabkan *overload* justru menjadi rugi – rugi daya yang terbuang. Hal seperti ini dapat mengindikasikan terjadinya pemborosan biaya. Terakhir adalah dengan meningkatnya arus rms akibat harmonik menyebabkan rele arus sering bekerja sebelum waktunya. Hal ini berdampak pada pemadaman dan gangguan kontinuitas suplai listrik ke konsumen. Sehingga para konsumen listrik khususnya di ranah bisnis akan mengalami kerugian ekonomi akibat terputusnya suplai listrik sehingga kegiatan industri menjadi terhenti [2].

## **2.2 Analisa Harmonisa [3]**

Analisis harmonik dapat dilakukan dengan berbagai cara, umumnya dilakukan dengan metode transformasi yakni mengubah atau mentransformasikan sinyal arus dari domain waktu ke domain frekuensi. Hal ini dikarenakan untuk melakukan pengamatan sinyal frekuensi yang tinggi, perlu diamati dalam domain frekuensi. Setelah dilakukannya proses transformasi, selanjutnya dilakukan proses perhitungan magnitude dan fasa dari komponen dasar maupun komponen – komponen harmonik pada orde yang lebih tinggi dari suatu gelombang.

### **2.2.1 Pengenalan Transformasi *Fourier***

Dalam menampilkan sinyal frekuensi perlu melalui transformasi sinyal dari domain waktu ke domain frekuensi karena terdapat frekuensi yang sangat tinggi, di dalam domain waktu kesulitan untuk menganalisis frekuensinya. Hal ini sangat penting dalam penganalisaan dalam

berbagai bidang yaitu telekomunikasi. Metode transformasi yang paling banyak dikembangkan untuk mendesain sistem pendeteksi harmonik dengan metode transformasi fourier beserta turunan dan pengembangannya.

### 2.2.2 Pengembangan Algoritma DFT

Dalam sistem waktu diskrit, transformasi sering dikenal *Discrete Fourier Transform* (DFT). Transformasi fourier

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt. \quad \dots\dots\dots(1)$$

Invers Transformasi:

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} df. \quad \dots\dots\dots(2)$$

Agar transformasi fourier dapat digunakan dalam operasi digital, maka diperlukan sampel-sampel pada kawasan frekuensi dan waktu. Sampel-sampel sinyal kontinyu pada kawasan waktu akan merepresentasikan keseluruhan sinyal kontintyu tersebut. Sampel-sampel ini akan mengubah sinyal kontinyu menjadi sinyal diskrit, maka dibutuhkan *Discrete Fourier Transform* (DFT). Transformasi fourier sinyal diskrit adalah sebagai berikut:

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn} \quad \dots\dots\dots(3)$$

Dengan

$$W_N = e^{-j(2\pi/N)}$$

Invers Transformasi :

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]W_N^{-kn} \quad \dots\dots\dots(4)$$

Pengambilan sampel untuk analisis DFT dari sinyal kontinyu perlu diperhatikan agar tidak terjadi kesalahan. Analisis DFT dari sinyal kontinyu menggunakan perkiraan berupa sampel-sampel, maka perlu dipahami adanya keterbatasan sampel-sampel terhadap bentuk sinyal

kontinyu yang sebenarnya, ada tiga yang bisa terjadi akibat kesalahan perkiraan sinyal kontinyu, yaitu:

- a. *Aliasing*, karena sample rate tidak cukup tinggi untuk menghindari overlap spektrum.
- b. *Leakage*, timbul efek distorsi spektrum karena pengabaian sinyal frekuensi pada waktu yang tak terhingga.
- c. *Picket-fence effect*, timbul karena ketidakmampuan DFT mengobservasi sinyal sebagai sinyal kontinyu, karena perhitungan spektrum yang terbatas.

### 2.3 Indikator dan Standar Distorsi Harmonik Arus

Indikator harmonik yang umum digunakan adalah THD (*Total Harmonic Distortion*) yang berelasi dengan gelombang tegangan. THD didefinisikan sebagai presentase nilai *root mean square* (r.m.s) dari komponen – komponen harmonik terhadap komponen dasar sesuai dengan persamaan berikut,

$$THD = \frac{\sqrt{\sum_{n=2}^N V_n^2}}{V_1} \dots\dots\dots(5)$$

- $V_n$  adalah besar tegangan harmonik orde n
- N adalah orde harmonik tertinggi yang dianalisis
- $V_1$  adalah tegangan fasa r.m.s pada frekuensi dasar

### 2.4 Mikrokontroler AVR [4]

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

Berbeda dengan *CPU* serbaguna, mikrokontroler tidak selalu memerlukan memori eksternal, sehingga mikrokontroler dapat dibuat lebih murah dalam kemasan yang lebih kecil dengan jumlah pin yang lebih sedikit, kecepatan eksekusi dalam 1 siklus *clock* dibandingkan dengan MCS51 yang membutuhkan 12 siklus *clock*, dapat digunakan untuk berbagai macam aplikasi. Mikrokontroler juga digunakan untuk mengolah data – data biner (*digital*) yang merupakan gabungan dalam bentuk suatu *chip* (IC) serta umunya terdiri dari alamat (*address*), data, pengendali, memori (RAM atau ROM), dan bagian *input-output*.



AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel dan berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 register *general-purpose*, *timer/counter* fleksibel dengan mode *compare*, *interrupt* internal dan eksternal, serial UART, *programabel Watchdog Timer*, *mode power saving*, ADC, dan PWM internal. AVR juga mempunyai *In-System Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan SPI.

#### 2.4.1 Mikrokontroler At-Mega 16

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) merupakan pengontrol utama standar industri dengan berbagai kelebihan yang dimiliki dibandingkan dengan mikroprosesor, yaitu murah, dukungan *software* dan dokumentasi yang memadai, dan membutuhkan komponen pendukung yang sangat sedikit. Salah satu tipe mikrokontroler AVR untuk aplikasi standar yang memiliki fitur memuaskan adalah ATmega16.

Pada umumnya mikrokontroler terdiri dari bagian-bagian sebagai berikut: alamat (*address*), data, pengendali, memori (*RAM atau ROM*), dan bagian *input-output*.

Spesifikasi Mikrokontroler AVR ATmega16

Fitur-fitur pada mikrokontroler ATmega16 antara lain:

- a. Saluran I/O ada 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
- b. ADC 10 *bit* sebanyak 8 *channel*.
- c. CPU yang terdiri dari 32 buah *register*.
- d. *Watchdog Timer* dengan osilator *internal*
- e. Dan fitur-fitur lain yang mempermudah dalam penggunaan.
- f. Tegangan kerja berkisar 4-5 V.
- g. Memori Flash 8 *Kbytes* untuk program
- h. Memori EEPROM 512 *bytes* untuk data
- i. Memori SRAM 512 *bytes* untuk data
- j. 20 *interrupt*
- k. Satu 16-bit *timer* dan dua 8-bit *timer*
- l. Komunikasi serial melalui SPI dan USART
- m. *Analog komparator*
- n. 4 I/O PWM
- o. Fasilitas *In System Programming (ISP)*

## 2.5 Sensor Arus ACS 712 05 [5]

Sensor arus yang digunakan adalah *DT-Sense Current Sensor* dengan kapasitas 5 A merupakan suatu modul sensor arus yang menggunakan IC sensor arus linier berbasis *Hall-Effect ACS712* produksi Allegro. Sensor arus ini dapat digunakan untuk mengukur arus AC atau DC. Untuk modul DT-SENSE dengan tipe *with OpAmp*, telah ditambahkan rangkaian *OpAmp* sehingga sensitivitas pengukuran arus dapat lebih ditingkatkan dan dapat mengukur perubahan arus yang lebih kecil. Sensor ini digunakan pada aplikasi-aplikasi di bidang industri, komersial, maupun komunikasi. Contoh aplikasinya antara lain untuk sensor kontrol motor, deteksi dan manajemen penggunaan daya, sensor untuk *switch-mode power supply*, sensor proteksi terhadap *overcurrent*, dan lain sebagainya. Gambar 2.3 berikut merupakan tampilan modul sensor arus ACS 712 buatan *Inovative Electronics*.



**Gambar 2.3** Sensor Arus ACS 712

Spesifikasi :

1. Berbasis ACS712 dengan fitur:
  - a. *Rise time output* = 5  $\mu$ s.
  - b. *Bandwidth* sampai dengan 80 kHz.
  - c. Total kesalahan *output* 1,5% pada suhu kerja TA = 25°C.
  - d. Tahanan konduktor internal 1,2 m $\Omega$ .
  - e. Tegangan isolasi minimum 2,1 kVRMS antara pin 1-4 dan pin 5-8.
  - f. Sensitivitas *output* 185 mV/A.
  - g. Mampu mengukur arus AC atau DC hingga 5 A.
  - h. Tegangan *output* proporsional terhadap input arus AC atau DC.
2. Tegangan kerja 5 VDC.
3. Dilengkapi dengan *OpAmp* untuk menambah sensitivitas *output* (untuk tipe *With OpAmp*).

Berikut adalah Tabel alokasi pin *interface* yang ditunjukkan pada Tabel 2.1 :

Tabel 2.1 Alokasi pin *Interface*

Input	Nama	Fungsi
1	VCC	Tegangan <i>Input</i> 5 VDC
2	<i>Out</i>	Tegangan keluaran sensor
3	<i>Out Amp</i>	Tegangan keluaran <i>Op Amp</i>
4	<i>Ground</i>	Titik referensi <i>Ground</i>

Tegangan *output* ACS712 terhubung ke pin *Out* dan tegangan *output* rangkaian *OpAmp* terhubung ke *Out\_Amp*. Rumus tegangan pada pin *Out* =  $2,5 \pm (0,185 \times I)$  Volt. Dimana I = arus yang terdeteksi dalam satuan Ampere. Rangkaian *OpAmp* yang dapat digunakan untuk meningkatkan sensitivitas dan mengubah *offset* pada tegangan output *OpAmp* (pin *Out\_Amp*). Sensitivitas atau *Gain* diatur melalui VR1 sedangkan *offset* diatur melalui VR2.

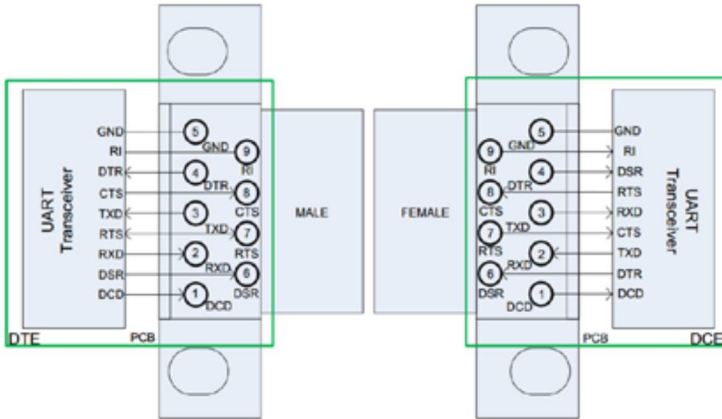
Pengujian sederhana dapat dilakukan dengan langkah-langkahnya sebagai berikut:

1. Hubungkan VCC (pin 1) pada modul DT-SENSE dengan sumber tegangan 5 Volt dan pin 4 dengan ground.
2. Ukur tegangan *Out* (pin 2) pada modul DT-SENSE dengan voltmeter.
3. Jika tidak ada arus yang lewat melalui input, maka *Out* akan bernilai sekitar 2,5 Volt.
4. Jika modul *DT-SENSE CURRENT SENSOR* dilengkapi dengan *OpAmp*, maka atur Offset (VR2) sehingga nilai *Out\_Amp* menjadi sekitar 2,5 Volt.
5. Lewatkan arus listrik melalui jalur input. Semakin besar arus yang melewati input, semakin besar pula perubahan tegangan pada pin *Out* dan pin *Out\_Amp*.
6. Karena sifat dari rangkaian *OpAmp* yang digunakan, jika tegangan *Out* lebih kecil dari 2,5 Volt, maka tegangan *Out\_Amp* akan lebih besar dari pada 2,5 Volt.
7. Sebaliknya, jika tegangan *Out* lebih besar dari 2,5 Volt, maka tegangan *Out\_Amp* akan lebih kecil dari pada 2,5 Volt.

## 2.6 Komunikasi *Serial RS-232* [6]

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

Port serial RS-232 yang umum digunakan adalah DB-9. Konfigurasi pin dan nama konektor DB-9 dapat dilihat pada Gambar 2.4 Keterangan tentang pin DB-9 terdapat pada Tabel 2.2



**Gambar 2.4** Konfigurasi Pin DB-9

**Tabel 2.2** Keterangan Pin DB-9

Singkatan	Keterangan	Fungsi
TD	Transmit Data	Untuk pengiriman data serial ( TD <sub>X</sub> )
RD	Receive Data	Untuk penerimaan data serial ( RD <sub>X</sub> )
RTS	Request To Send	Sinyal untuk menginformasikan perangkat bahwa UART siap melakukan pertukaran data
CTS	Clear To Send	Digunakan untuk memberitahukan bahwa perangkat siap untuk melakukan pertukaran data
DSR	Data Set Ready	Memberitahukan UART bahwa perangkat siap untuk melakukan pertukaran data
SG	Signal Ground	Dihubungkan ke ground
CD	Carrier Detect	Saat perangkat mendeteksi suatu carier, dari perangkat lain, maka sinyal ini akan di aktifkan
DTR	Data Terminal Ready	Untuk memberitahukan bahwa UART siap melakukan pertukaran data
RI	Ring Indikator	Akan aktif jika ada sinyal masuk

## 2.7 Pemrograman Grafis Menggunakan Delphi [7]

*Delphi* merupakan Pemrograman Terintegrasi (*Integrate Development Environment / IDE*). *Delphi* bukan bahasa pemrograman, tetapi perangkat lunak yang menyediakan seperangkat alat (*tools*) untuk membantu pemrogram dalam menulis program komputer. *Delphi* menggunakan *Object Pascal* sebagai bahasa pemrogramannya. *Object Pascal* merupakan bahasa *Pascal* yang diberi tambahan kemampuan untuk menerapkan konsep - konsep *OOP (Object Oriented Programming)*. Seluruh sintak *Object Pascal* menggunakan aturan yang ada di dalam *Pascal*, termasuk perintah-perintah dasar seperti *control structures, variabels, array*, dan sebagainya.

Peralatan yang disediakan oleh *Delphi* memberikan kemudahan bagi pemrogram untuk membuat program secara *visual (visual programming)*. *Visual programming* adalah metoda dimana sebagian atau keseluruhan program dibuat dengan cara menggambarkan tampilan / hasil akhir dan kemudian meminta beberapa perangkat untuk membuat kode-kode program berdasarkan gambaran hasil akhir tersebut.

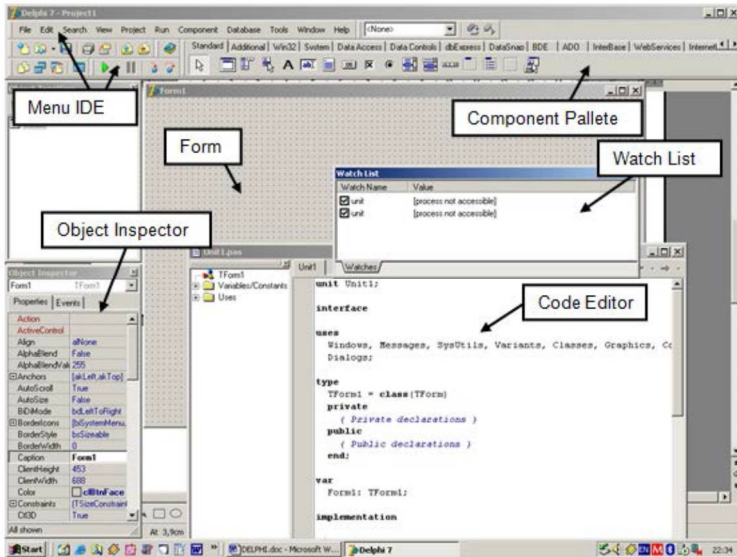
Karena program yang dibuat di dalam *Delphi* berjalan di dalam sistem operasi *Windows* maka kegiatan program dilakukan berdasarkan metoda *event-driven programs*. *Event-driven programming* adalah metoda mengeksekusi kode program berdasarkan pesan (*messages / events*) yang diberikan oleh pemakai ataupun oleh sistem operasi atau program lainnya. Sebagai contoh : apabila pemakai menekan tombol kiri *mouse* dan kemudian melepaskannya dengan cepat (kita mengenal itu sebagai klik) maka tindakan tersebut akan membuat aplikasi menerima pesan *mouse down* yang disertai dengan informasi tombol mana yang ditekan dan lokasi kursor saat klik dilakukan, tetapi apabila pemakai menekan tombol kiri *mouse* dan kemudian menggeser *mouse* tanpa melepaskan tombol kiri maka aplikasi akan menerima pesan *mouse move*.

### 2.7.1 Peralatan pada Delphi

Beberapa peralatan yang disediakan oleh *Delphi* dan cukup diketahui antara lain:

- a. *Code Editor*
- b. *Form*
- c. *Object Inspector*
- d. *Component Palette*
- e. *Project Manager*
- f. *Watch List*
- i. dan sebagainya

Gambar 2.5 menunjukkan Tampilan awal pada *Delphi* dan beberapa peralatan yang terdapat pada toolbar. Peralatan-peralatan tersebut digunakan sebagai sarana untuk memasukkan program ataupun membuat tampilan untuk aplikasi yang akan dibuat.



**Gambar 2.5** Tampilan pada *Software Delphi*

### 2.7.2 Code Editor

*Code Editor* merupakan peralatan yang digunakan untuk menuliskan kode-kode program. *Code Editor* menyediakan sejumlah fasilitas penyuntingan (*editing*) seperti : *copy*, *cut*, *paste*, *find*, *replace*, dan sebagainya. *Code Editor* mengetahui apakah yang ditulis merupakan perintah *Object Pascal* atau bukan dan menampilkan tulisan sesuai dengan tipe / kelompok tulisan tersebut.

### 2.7.3 Form

*Form* merupakan area dimana pemrogram meletakkan komponen-komponen *input* dan *output*. *Delphi* akan secara otomatis membuat kode-kode program untuk membuat dan mengatur komponen-komponen tersebut. Umumnya pada setiap aplikasi ada paling tidak satu buah *form* dan *form* tersebut dijadikan sebagai *form* utama (*Main Form*).

Setiap *form* selalu disimpan dalam 2 buah file, yaitu : 1) file dengan akhiran *.dfm* ; 2) file dengan akhiran *.pas*. File berakhiran *.dfm* menyimpan informasi mengenai komponen-komponen yang ada di

dalam *form* sedangkan file berakhiran *.pas* menyimpan informasi mengenai kode-kode program yang berhubungan dengan *form* tersebut. Tetapi, tidak setiap file *.pas* selalu mempunyai pasangan *.dfm*.

#### **2.7.4 Component Palette**

*Component Palette* adalah peralatan yang menyediakan daftar komponen yang dapat digunakan oleh pemrogram. Komponen di dalam *Delphi* dibedakan menjadi dua macam, yaitu :

1. Komponen Visual
2. Komponen Non Visual

Komponen Visual adalah komponen yang memberikan tampilan tertentu pada saat dimasukkan ke dalam *form*, sedangkan komponen non-visual adalah komponen yang tidak memberikan tampilan tertentu saat dimasukkan ke dalam *form*. Komponen non-visual yang dimasukkan ke dalam *form* hanya ditampilkan sebagai sebuah kotak berisi simbol tertentu.

#### **2.7.5 Object Inspector**

*Object Inspector* adalah peralatan yang digunakan untuk mengatur properti dari komponen yang ada di *form* termasuk properti *form*. *Object Inspector* memberi dua macam peralatan, yaitu :

1. *Properties*
2. *Events*

Peralatan *Properties* adalah peralatan yang digunakan untuk mengubah atau mengatur nilai-nilai dari properti komponen sedangkan Peralatan *Events* digunakan untuk membuat *event-handler*. *Event handler* adalah prosedur yang digunakan khusus untuk menanggapi satu *event / message* tertentu.

#### **2.7.6 Watch List**

*Watch List* merupakan peralatan yang digunakan untuk memeriksa isi satu variabel atau properti tertentu saat program sedang dieksekusi. *Watch List* biasanya digunakan bersamaan dengan *Break Points* dan *Step-by-Step execution*. Kita akan mempelajari ini pada akhir pelatihan untuk melihat bagaimana mencari kesalahan di dalam program dengan cepat.

#### **2.7.7 Tombol-tombol ShortCut**

Untuk berpindah dari satu peralatan ke peralatan lain anda dapat melakukannya dengan mengklik jendela dari peralatan tersebut, tetapi ada cara yang lebih cepat untuk berpindah dari satu peralatan ke

peralatan lain. Tabel 2.3 menunjukkan tombol *keyboard* yang dapat digunakan untuk keperluan tersebut.

**Tabel 2.3** Tombol *Shortcut*

No	Tombol <i>Keyboard</i>	Keterangan
1	F12	Pindah dari <i>Form</i> ke <i>Code Editor</i> dan sebaliknya
2	F11	Pindah ke <i>Object Inspector</i>
3	F10	Pindah ke menu utama
4	<i>Shift+F5</i>	Menampilkan <i>Watch List</i>



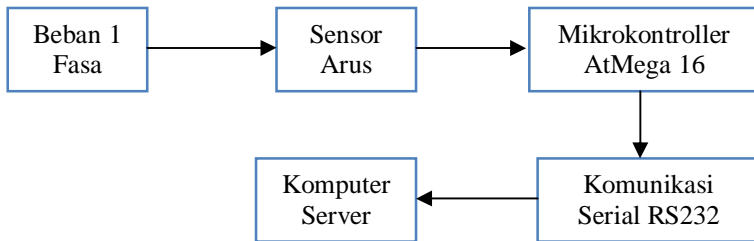
*Halaman ini sengaja dikosongkan*

## BAB III PERANCANGAN ALAT

Perancangan sistem pendeteksian dan *monitoring* harmonik arus ini dibagi menjadi tiga tahap, diantaranya perencanaan, pembuatan modul pendeteksi, pembuatan *software*, analisis dan *monitoring*.

Pada tahap pertama yakni tahap perencanaan meliputi pengujian komponen yang digunakan, khususnya pengujian sensor arus. Pada penelitian ini digunakan sensor ACS 712 dengan arus nominal 5 A. Pengujian ini dilakukan untuk menentukan spesifikasi dari komponen yang dipakai dalam tugas akhir ini sebagai acuan untuk mendesain akuisisi data dan program analisis.

Tahap kedua adalah pembuatan modul pendeteksi. Pada tahap ini dilakukan perakitan beberapa komponen sebagai modul pendeteksi, diantaranya mikrokontroler ATMEGA 16, kabel RS232, dan *Personal Computer*. Beberapa komponen tersebut dirakit menjadi satu yang digunakan untuk mendeteksi, mengambil sampel dan melakukan akuisisi data gelombang arus dengan personal computer. Berikut diagram *blok* modul pendeteksi yang disajikan dalam gambar 3.1.



**Gambar 3.1** Diagram *blok* modul pendeteksian

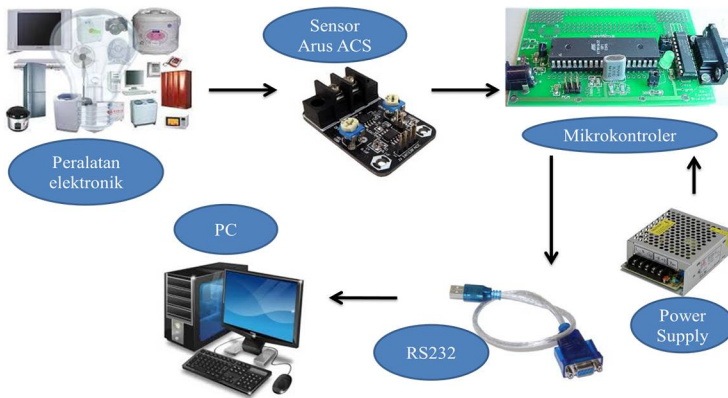
Tahap ketiga adalah pembuatan *software* analisis dan *monitoring*. Pada tahap ini dibuat program untuk melaksanakan beberapa fungsi yaitu pengambilan data dari modul pendeteksi, pengkondisian sinyal masukan dimana dari nilai tegangan dikembalikan ke nilai arus, analisis harmonik arus, serta menampilkan dan menyimpan data secara. Pembuatan program dilakukan dengan menggunakan bahasa pemrograman grafis *Borland delphi*. Berikut diagram *blok* dari *software* analisis dan *monitoring* harmonik arus yang tersaji dalam gambar 3.2,

### 3.1 Perancangan dan Pembuatan *Hardware*

Dalam proses pembuatan *hardware* pada monitoring harmonisa arus listrik menggunakan mikrokontroler dan terdiri dari beberapa peralatan elektronik untuk menunjang kerjanya. Beberapa peralatan tersebut terdiri dari modul mikrokontroler, modul sensor arus (ACS 712), *downloader*, rangkaian konverter serial ke ethernet WIZ100SR, *power supply*, dan papan beban. Pada perancangan alat, dibuat hanya untuk memonitoring 1 fasa, sehingga hanya menggunakan 1 sensor arus saja.

Untuk mendapatkan nilai arus yang bermacam-macam digunakan beban yang berbeda-beda, diantaranya menggunakan lampu pijar dan beberapa peralatan elektronik yang sering digunakan dalam keseharian.

Untuk lebih rincinya dapat dilihat pada gambar 3.2 berikut mengenai diagram fungsional alur kerja dari alat monitoring harmonisa arus listrik :



**Gambar 3.2** Diagram Fungsional *Hardware*

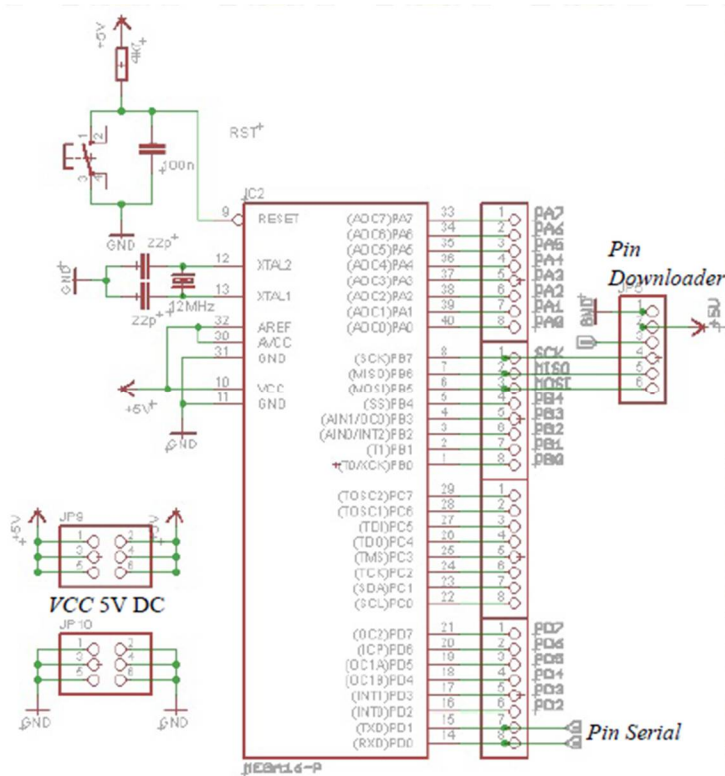
#### 3.1.1 Rangkaian Mikrokontroler

Rangkaian mikrokontroler adalah rangkaian kompleks yang terdiri dari IC mikrokontroler, *supply*, *reset*, dan *osilator* eksternal. Rangkaian mikrokontroler ini bisa dihubungkan dengan rangkain lain untuk menjalankan fungsi tertentu. *Port-port* pada mikrokontroler ini dihubungkan dengan rangkaian sensor arus dan rangkaian komunikasi *serial* RS-232. Skematik rangkaian minimum system dapat dilihat pada Gambar 3.3. Skematik rangkaian mikrokontroler menunjukkan bahwa *port-port* pada *AT-Mega16* difungsikan sesuai kebutuhan. Pada *portA*

digunakan untuk data *input* dari sensor arus dan *port D* digunakan untuk komunikasi serial, *driver* kontaktor.

### Input ADC

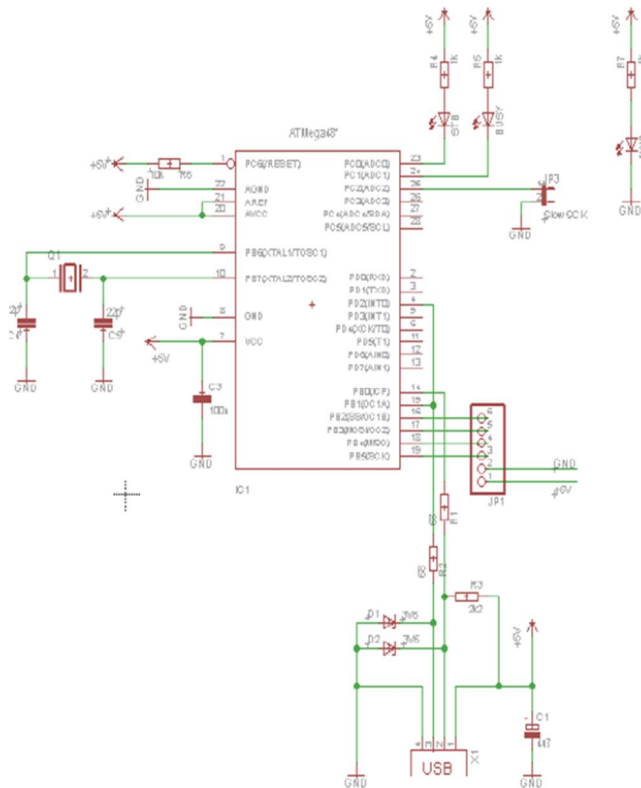
*PortA* pada *AT-Mega16* pada Tugas Akhir ini difungsikan sebagai *port* penerima data sensor. Data dari sensor arus yang terukur akan masuk *port* ADC tadi dalam besaran tegangan sebesar 0 s.d 5 Volt DC.



Gambar 3.3 Skematik rangkaian mikrokontroler

### 3.1.2 Downloader

*Downloader* merupakan komponen yang digunakan untuk *transfer*-kan program dari *Code Vision AVR* kedalam mikrokontroler. Tanpa adanya *downloader* ini, maka program tidak dapat dimasukkan pada mikrokontroler. Gambar 3.4 berikut merupakan rangkaian *downloader* yang digunakan



**Gambar 3.4** Skematik rangkaian downloader

Downloader ini menggunakan IC mikrokontroler At-Mega 8. Berbeda dengan At-Mega 16 yang digunakan pada modul mikrokontroler diatas. At-Mega 8 ini memiliki I/O yang lebih sedikit, namun fitur-fitur didalamnya sama dengan At-Mega 16 hanya saja jumlahnya yang lebih sedikit.

### 3.1.3 Komunikasi Serial RS-232 ke PC

Komunikasi serial tugas akhir ini menggunakan sistem RS-232 dan menggunakan IC Max 232. Komunikasi serial digunakan untuk mengirim hasil pengolahan data ke PC yang akan ditampilkan ke tampilan Delphi. Komunikasi tersebut memerlukan kabel penghubung

antara rangkaian RS-232 dengan PC yaitu kabel *serial to USB* seperti pada Gambar 3.5.

IC *MAX 232* yang digunakan mempunyai 16 kaki dengan *supply* tegangan sebesar 5 volt. Kaki ke-16 sebagai *input* tegangan (*Vcc*), kaki ke-15 sebagai *ground* (*GND*). Kaki 8 dan 13 sebagai *input* RS-232, sedangkan kaki 7 dan 14 sebagai *output* RS-232.



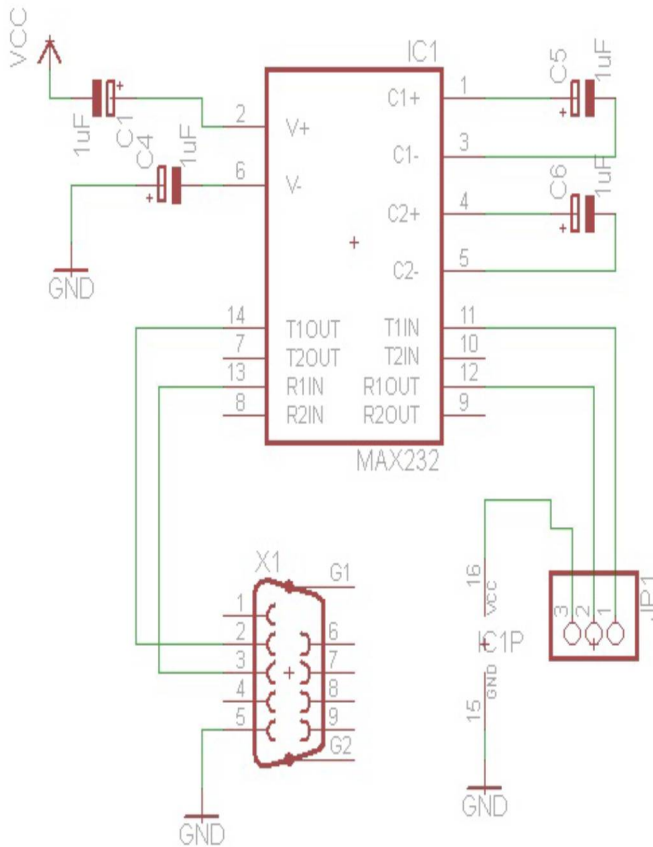
**Gambar 3.5** Kabel *serial* RS-232

RS-232 merupakan suatu *interface* yang menghubungkan antara terminal data dari suatu peralatan yang melakukan pertukaran data *biner* secara *serial*. Adapun karakteristik dari RS-232 dapat dilihat pada Tabel 3.1.

**Tabel 3.1** Karakteristik *Serial* RS-232

<b>Karakteristik</b>	<b>RS-232</b>
Max driver/receiver	1/1
Mode	<i>Half &amp; full</i>
Topologi	<i>Point to point</i>
Jarak	15m
<i>Speed</i> (tiap12m)	20kbs
Logika "1"	-3V s/d -25V
Logika "0"	+3V s/d +25V

Daerah tegangan antara -3 Volt hingga +3 Volt adalah *invalid* level, yaitu daerah tegangan yang tidak memiliki level logika pasti sehingga harus dihindari. Demikian juga, level tegangan lebih negatif dari -25 Volt atau lebih positif dari +25 Volt juga harus dihindari karena tegangan tersebut dapat merusak *line driver* pada saluran RS-232. Untuk mendapatkan komunikasi *serial* diperlukan skematik rangkaian yang dapat dilihat pada Gambar 3.6.

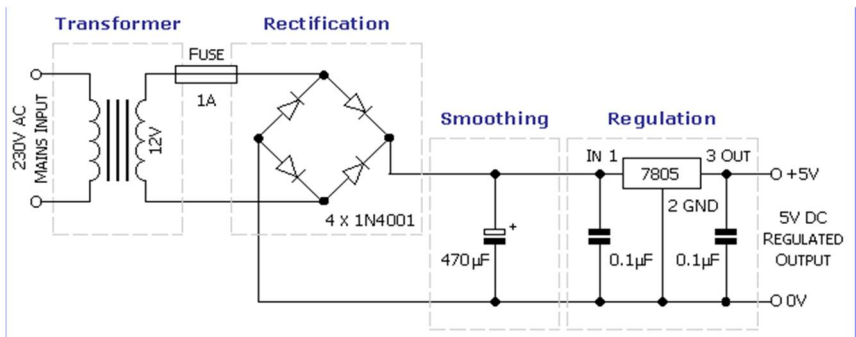


**Gambar 3.6** Skematik rangkaian komunikasi serial

### 3.1.4 Power Supply

*Power supply* digunakan untuk men-supply komponen-komponen yang memerlukan tegangan 5 Vdc atau lebih. *Power supply* pada alat ini digunakan untuk men-supply modul mikrokontroler, sensor arus, dan rangkaian komunikasi.

Cara kerja dari *power supply* ini menurunkan tegangan dari PLN 220 Volt menggunakan trafo *step down* yang kemudian akan disearahkan menggunakan rangkaian dioda *bridge* dan di *filter* dengan kapasitor dan resistor untuk menghilangkan *ripple* yang terjadi. Rangkaian power supply ditunjukkan pada gambar 3.7.



**Gambar 3.7** Rangkaian power supply

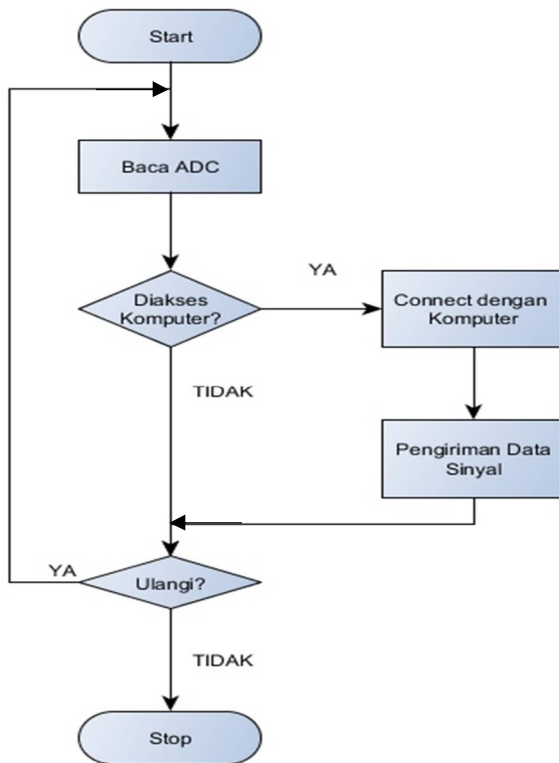
### 3.2 Perancangan Pembuatan *Software*

Dalam pembuatan *software* analisis dan *monitoring* dibagi menjadi tiga bagian yakni pengkondisian sinyal yang diterima dari mikrokontroler, analisis harmonik menggunakan transformasi – s, dan penentuan nilai THDi dari sinyal arus yang telah diperoleh. Pengkondisian sinyal pada *visual basic* dilakukan guna memperoleh nilai arus yang sebenarnya, karena dalam proses pengiriman sinyal arus yang sebenarnya telah mengalami beberapa konversi sebelum diterima oleh *software* analisis dan *monitoring*. Setelah proses rekonstruksi kembali selesai maka diperoleh sinyal arus yang sebenarnya, sinyal arus inilah yang berikutnya dianalisis dengan menggunakan metode transformasi – s guna mengetahui gelombang – gelombang lain yang terbentuk di luar frekuensi fundamental nya. Setelah diketahui nilai amplitudo dengan nilai frekuensinya barulah langkah berikutnya adalah menentukan nilai THDi dari sinyal arus tersebut. Hasil olah data tersebut kemudian dibawa ke perhitungan Matlab menggunakan *fuzzy logic* untuk dilakukan identifikasi pada peralatan.

#### 3.2.1 Perancangan Program dengan *Code Vision AVR*

Perancangan program pada mikrokontroler Atmega 16 menggunakan *software* aplikasi *code vision AVR* yang menggunakan bahasa C sebagai programnya. Dalam program, mikrokontroler mengolah data ADC yang berupa nilai arus *input* dan dimasukkan dalam rumus, jika nilai dari perhitungan memenuhi syarat tertentu langsung mengirim data nilai arus dan tegangan. Gambar 3.8 merupakan *Flow Chart* dari program mikrokontroler :





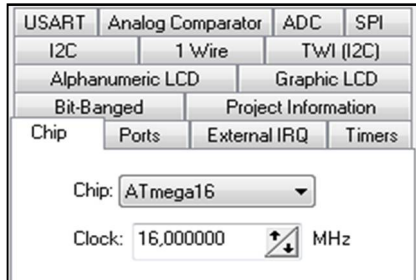
**Gambar 3.8** Flowchart code vision AVR

Sebelum melakukan pembuatan program ada beberapa hal yang harus diatur pada *Code Vision AVR* agar sesuai dengan *hardware* yang telah kita buat. Berikut merupakan hal-hal yang harus di *setting* terlebih dahulu :

1. *Setting* Chip At-Mega 16

Langkah pertama dalam mendesain ADC adalah menentukan chip yang digunakan seperti yang ditunjukkan pada gambar 3.9. Atau bisa dikatakan pemilihan mikrokontroler yang digunakan untuk pendeteksi serta dilakukannya proses ADC. Dalam pemilihan chip, juga harus memperhatikan spesifikasi dari chip tersebut sehingga dapat diketahui kemampuan dari mikrokontroler jenis tersebut. Setelah dilakukan pemilihan chip, berikutnya dilakukan pemilihan clock eksternal atau

yang lebih dikenal dengan kristal *clock*. Semakin besar kristal *clock* yang dipilih, semakin besar pula kemampuan *baudrate* atau frekuensi *sampling* dari mikrokontroler tersebut.

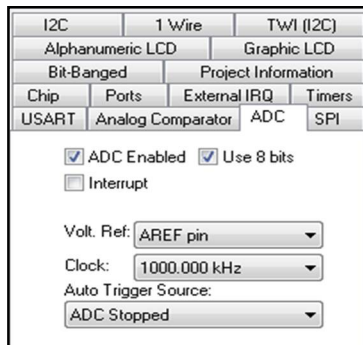


**Gambar 3.9** Pengaturan *chip* dan *clock*

## 2. Mengatur Jumlah Bit

Pada pengaturan jumlah bit menggunakan 8 bit ADC yang ditunjukkan pada gambar 3.10 karena *default* protokol komunikasi serial adalah 8 bit ADC yang sudah disesuaikan dengan *start* bit dan *stop* bit.

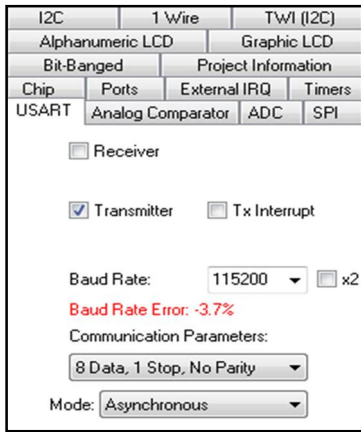
Pada *sampling rate* dipilih maksimum 1000 KHz supaya akurasi dari proses *sampling* sinyal mempunyai *error* yang sedikit. Dimana *sampling rate* ini merupakan batas frekuensi yang mampu dikirim setiap detiknya oleh ADC mikrokontroler. Dimana setiap konversi data membutuhkan 10 *clock*, sehingga frekuensi yang mampu dikonversi adalah 75 KHz. Sebenarnya *sampling rate* dapat lebih besar dari pada nilai tersebut, akan tetapi 1000 KHz sudah menjadi nilai maksimum karena efek dari pemilihan kristal *clock* yang cukup besar.



**Gambar 3.10** Pengaturan jumlah *bit* ADC serta *sampling rate* dengan kristal *clock* 16 MHz

### 3. Setting Pengiriman Data

Dalam Tugas Akhir menggunakan *system* pengiriman melalui komunikasi serial RS232, jadi dalam *codevison* harus di-*setting* program pengiriman (*transmitter*) di dalam mikrokontroler agar dapat mengirim data yang terbaca oleh ADC. Gambar 3.11 merupakan *setting* yang dilakukan pada *Code Vision AVR* :



**Gambar 3.11** Pengaturan Komunikasi Serial

### 3.2.2 Pembuatan Software Analisis Harmonisa dan Monitoring

Pada analisis harmonisa ini, input arus yang didapatkan pada pengkondisian sinyal akan disimpan pada suatu file dalam domain waktu tertentu dan jumlah sampel tertentu.

Tujuan penyimpanan file arus ini digunakan untuk mengubah ke dalam domain frekuensi menggunakan *Discrete Fourier Transform (DFT)*.

*Discrete Fourier Transform (DFT)* merupakan prosedur matematika yang digunakan untuk menentukan harmonisa. Persamaan *Discrete Fourier Transform (DFT)* dapat dilihat pada persamaan 3.1 dibawah ini yaitu sebagai berikut :

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \quad \dots\dots\dots (3.1)$$

Dimana  $x(n)$  adalah sekuens diskrit nilai sampel dalam fungsi waktu dari variable continue  $x(t)$ .

Persamaan 3.1 tersebut jika dinyatakan dalam bentuk rectangular didapatkan persamaan 3.2 dibawah ini.

$$X(m) = \sum_{n=0}^{N-1} x(n) \left[ \cos\left(\frac{2\pi nm}{N}\right) - j \sin\left(\frac{2\pi nm}{N}\right) \right] \dots\dots\dots(3.2)$$

Dimana :

X(m) = Komponen output DFT ke m (contohnya : X(0),X(1))

m = DFT dalam domain frekuensi (m = 0, 1, 2, 3, ..., N-1)

x(n) = sampel input, x(0), x(1), x(2), dan seterusnya

n = sampel input dalam fungsi waktu (n= 0, 1, ..., N-1)

J =  $\sqrt{-1}$  ( bilangan imajiner)

N = jumlah sampel input

Hasil dari *Discrete Fourier Transform (DFT)* ini digunakan untuk perhitungan dalam menentukan harmonisa setelah mendapatkan nilai frekuensi dasar dan frekuensi kelipatan dari frekuensi dasar. Kemudian dari hasil persamaan 3.2 tersebut dibuatlah program sesuai gambar 3.12.

```

var ax, bx, delta : real;
j,k,n, blok : integer;
begin
delta := 1/(2500*0.000664011);

for k:=0 to 1249 do
begin
ax:=0;bx:=0;
for j:=0 to 2499 do
begin
Xr[k]:=ax+x[j]*cos(2*3.14*k*j/2500);
Xi[k]:=bx-x[j]*sin(2*3.14*k*j/2500);
ax:=Xr[k]; bx:=Xi[k];
end;
Mx[k]:=sqrt((sqr(Xr[k])+sqr(Xi[k])))/2500;
Series4.AddXY(k*delta, Mx[k]);
end;

```

**Gambar 3.12** Listing program DFT

Nilai THDi didapatkan dari hasil perhitungan *Discrete Fourier Transform (DFT)* yang sudah di program pada gambar 3.10. Persamaan untuk mencari nilai THDi dapat dilihat pada persamaan 3.3 yang bisa dilihat dibawah ini sebagai berikut:

$$THD_i = \frac{\sqrt{I_2^2 + I_3^2 + I_4^2 + \dots + I_n^2}}{I_1} \dots\dots\dots(3.3)$$

Keterangan

$i_n$  adalah besarnya arus harmonisa orde n

N adalah orde harmonisa tertinggi yang dianalisa

$I_1$  adalah arus fasa pada frekuensi dasar

Dari rumus pada persamaan 3.3 diatas dijelaskan bahwa penjumlahan nilai kuadrat dari amplitudo arus pada kelipatan frekuensi fundamental 50 Hz dimasukkan dalam persamaan akar pangkat dua dibagi dengan amplitudo arus frekuensi fundamental . Konsep mencari THDi dari rumus pada persamaan 3.3 tersebut dibuat listing programnya yang bisa dilihat pada gambar 3.13 dibawah ini sebagai berikut.

```

delta := 1/(2500*0.000664011);
bag_atas:=0;
for abc:=1 to 6 do
begin
  nilai_max:=0;
  for scan_freq:= round(((2*abc)*50/delta)) to
round(((2*abc)*50/delta)) do
  begin
    if (Mx[scan_freq]>nilai_max) then
    begin
      nilai_max:=Mx[scan_freq];
    end;
  end;
  bag_atas:=bag_atas+sqr(nilai_max);
end;
nilai_max:=0;
for scan_freq:= round((50/delta)) to round((50/delta)) do
begin
  if (Mx[scan_freq]>nilai_max) then
  begin
    nilai_max:=Mx[scan_freq];
  end;
end;
nilai_THD:=sqr(bag_atas)/nilai_max*100;
edit3.Text:=floattostr(nilai_THD);
end;

```

**Gambar 3.13** Listing program mencari nilai THDi

Nilai THDi dan sinyal arus dari penelitian yang sudah dilakukan ditampilkan dalam view form. Pada view for ini saat pengukuran nilai yang ditampilkan terlebih dahulu adalah sinyal ADC dan sinyal Arus. Untuk nilai arus r.m.s, nilai DFT dan nilai THDi perlu disimpan dulu data sinyal arus dalam rentang waktu tertentu.

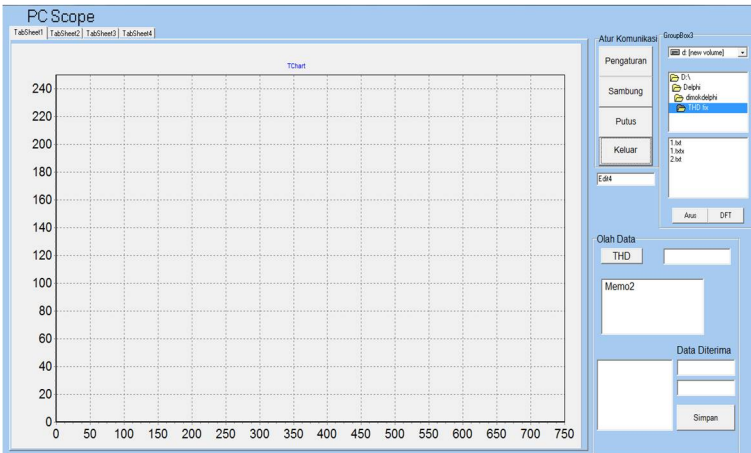
*Delphi 7* selain disebut sebagai bahasa pemrograman (*Language Program*), juga sering disebut sebagai sarana (*Tool*) untuk menghasilkan program-program aplikasi berbasis *Windows*. Secara umum ada beberapa manfaat yang di peroleh dari pemakaian program *Delphi 7*. Program *Delphi 7* berguna untuk membuat aplikasi beserta tampilan atau *Human Machine Interface*. Flowchart tampilan *Delphi 7* dapat dilihat pada Gambar 3.14.



**Gambar 3.14** Flowchart program *delphi*

## 1. Bagian *Monitoring*

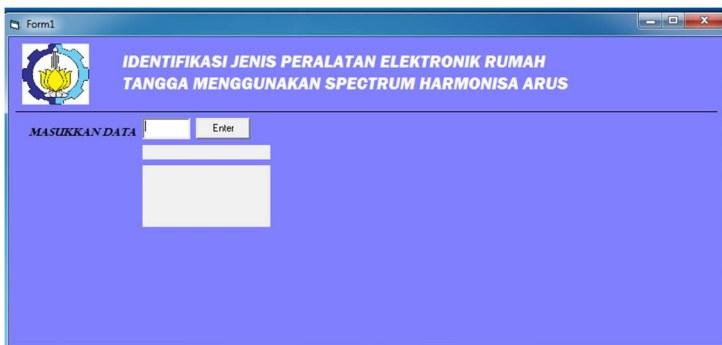
Bagian *monitoring* merupakan halaman utama dari aplikasi ini. Gambar 3.14 berikut merupakan tampilan *monitoring* harmonisa arus.



**Gambar 3.15** Tampilan *monitoring* harmonisa

## 2. Bagian Identifikasi

Pada bagian identifikasi di tugas akhir ini menggunakan *Visual Basic* dalam melakukan identifikasi harmonisa pada beban, sehingga dapat ditemukan harmonisa tiap peralatan rumah tangga. Pada gambar 3.16 merupakan program yang digunakan dalam identifikasi

The image shows a screenshot of a Visual Basic form titled "Form1". The form has a blue background and a title bar. At the top left, there is a logo featuring a gear and a lightbulb. To the right of the logo, the text reads "IDENTIFIKASI JENIS PERALATAN ELEKTRONIK RUMAH TANGGA MENGGUNAKAN SPECTRUM HARMONISA ARUS". Below the title, there is a section labeled "MASUKKAN DATA" with a text input field and an "Enter" button. Below the input field, there are two empty rectangular boxes, likely for displaying data or results.

**Gambar 3.16** Tampilan Identifikasi

## BAB IV PENGUJIAN DAN ANALISA DATA

Pada bab ini akan membahas mengenai hasil pengujian dan analisa atas penyusunan alat untuk “ Pengidentifikasi Harmonisa Pada Peralatan Rumah Tangga menggunakan Spektrum Harmonisa Arus”. Data pengujian sangat diperlukan untuk implementasi karena kinerja suatu sistem sangat dipengaruhi oleh kinerja per bagian dari sistem tersebut.

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.

Pengujian tersebut meliputi:

1. Pengujian tegangan *output power supply*
2. Pengujian rangkaian Mikrokontroler
3. Pengujian sensor arus
4. Pengujian komunikasi *serial* ke PC
5. Pengujian alat keseluruhan ke *Delphi 7*
6. Analisa menggunakan matlab.

### 4.1 Pengujian Tegangan *Output Power Supply*

Sumber tegangan dari rangkaian mikrokontroler, *driver* kontaktor dan kontaktor didapatkan dari rangkaian *power supply*. Pada rangkaian mikrokontroler dibutuhkan tegangan 5 V DC .

Untuk mengetahui apakah tegangan yang ada sesuai dengan yang kita inginkan, maka perlu adanya pengujian yaitu dengan hasil pada Tabel 4.1.

Persentase *error* diperoleh dengan rumus :

$$\% \text{ Error} = \frac{[\text{Nilai spesifikasi} - \text{Terukur (manual)}]}{\text{Nilai spesifikasi}} \times 100\% \dots\dots\dots (4.1)$$

Nilai spesifikasi

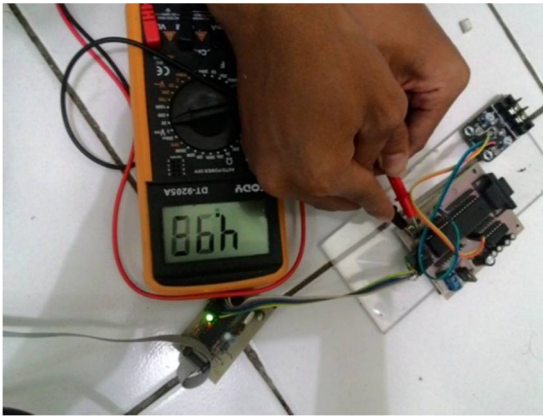
**Tabel 4.1** Hasil pengujian *power supply*

Rangkaian	Tegangan Spesifikasi (Volt)	Tegangan Terukur (Volt)	%Error
Power Supply	5 Volt	5 Volt	0



## 4.2 Pengujian Rangkaian Mikrokontroler

Dalam tahap ini, pengujian dilakukan dengan memberikan program sederhana dimana seluruh *port* pada ATmega16 diberikan logika 1 melalui program yang kemudian di-*download* ke mikrokontroler. Tegangan *supply* untuk rangkaian mikrokontroler adalah sebesar 5V. Pengukuran *output* pada mikrokontroler dilakukan dengan cara mengukur tegangan pada tiap-tiap kaki pada mikrokontroler terhadap *ground*. Cara pengujiannya adalah dengan menghubungkan *probe* merah dari *Multimeter* ke kaki-kaki mikrokontroler dan *probe* hitam ke *ground* seperti Gambar 4.1. Hasil pengujian rangkaian mikrokontroler pada Tabel 4.1.



Gambar 4.1 Pengujian mikrokontroler

Tabel 4.2 Pengujian mikrokontroler

Port A	Vout (Volt)	Port B	Vout (Volt)	Port C	Vout (Volt)	Port D	Vout (Volt)
0	4.99	0	4.99	0	4.99	0	4.99
1	4.99	1	4.99	1	4.99	1	4.99
2	4.99	2	4.99	2	4.99	2	4.99
3	4.99	3	4.99	3	4.99	3	4.99
4	4.99	4	4.99	4	4.99	4	4.99
5	4.99	5	4.99	5	4.99	5	4.99
6	4.99	6	4.99	6	4.99	6	4.99
7	4.99	7	4.99	7	4.99	7	4.99

Tegangan keluaran dari tiap-tiap *port* hampir sama dengan tegangan masukan ke rangkaian dengan *error* sebesar 0,2%, hal ini menunjukkan bahwa tiap-tiap *port* pada rangkaian *minimum* mikrokontroler masih dapat bekerja dengan baik.

### 4.3 Pengujian Sensor Arus

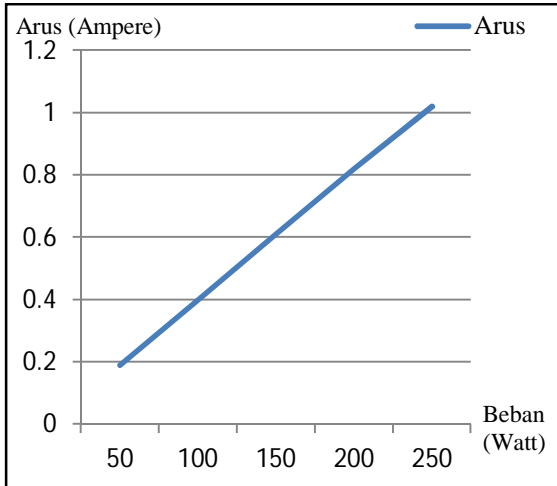
Sensor arus merupakan komponen yang digunakan untuk mendeteksi apakah terdapat arus yang mengalir pada sebuah sistem. Sensor yang digunakan merupakan sensor *DT-Sense ACS 712*.

Cara pengujian sensor arus yaitu dengan cara memberikan *input*-an tegangan dengan menggunakan berbagai macam beban. Data yang pertama menggunakan beban lampu pijar sebesar 25 Watt dan 100 Watt, data-data selanjutnya akan menggunakan beban seperti laptop, setrika, *heater*, dan kipas angin. Pada sisi *output* sensor tegangan *output*-nya diukur. Tabel 4.3 memperlihatkan hasil pengukuran arus dengan lampu pijar sebagai beban :

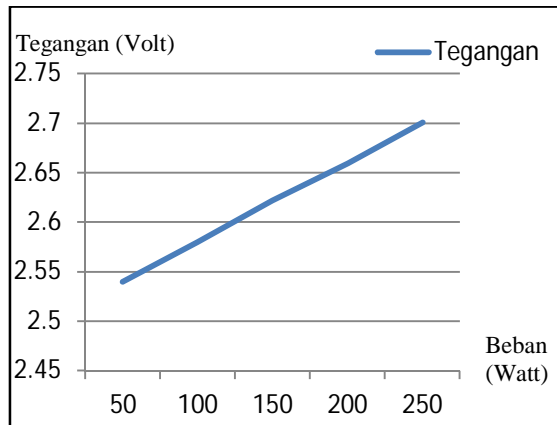
**Tabel 4.3** Data Pengujian Pertama Sensor Arus

Beban (Watt)	Arus (Ampere)	Tegangan <i>Output ACS 712</i> (Volt)	Tegangan Beban (Volt)
50	0,19	2,54	0,04
100	0,4	2,58	0,08
150	0,61	2,622	0,122
200	0,82	2,66	0,16
250	1,02	2,701	0,201

Dari data pada tabel 4.3 dapat dilihat linieritasnya arus dan tegangan *output ACS 712* terhadap perubahan beban melalui grafik pada gambar 4.2 dan gambar 4.3 berikut :



**Gambar 4.2** Grafik hasil arus dari sensor arus ACS 712



**Gambar 4.3** Grafik hasil tegangan *output* ACS 712

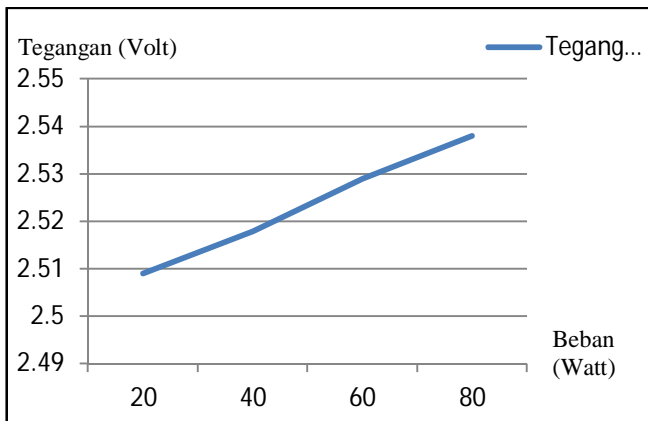
Untuk melihat keakurasian sensor arus yang digunakan maka pengambilan data dilakukan sebanyak dua kali. Data kedua menggunakan beban lampu LHE (Lampu Hemat Energi) sebanyak 4 buah @20 Watt dengan tegangan sumber sebesar 220 Volt. Tabel 4.4 berikut memperlihatkan hasil pengambilan data :

**Tabel 4.4** Data Pengujian Kedua Sensor Arus

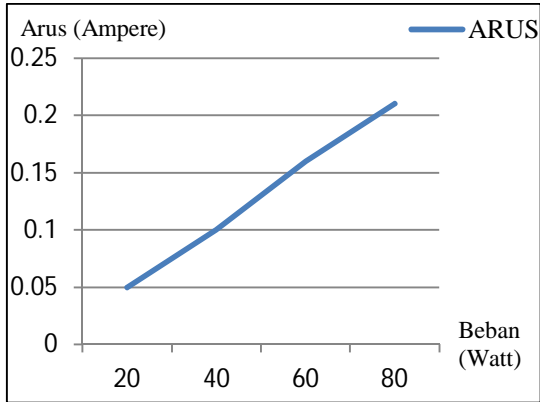
Beban (Watt)	Arus (Ampere)	Tegangan <i>Output</i> ACS 712 (Volt)	Tegangan Baban (volt)
20	0.05	2.509	0.009
40	0.10	2.518	0.018
60	0.16	2.529	0.029
80	0.21	2.538	0.038

Data yang didapatkan pada tabel 4.4 juga hampir sama dengan tabel 4.3. kenaikan arus dibarengi dengan kenaikan tegangan *output* sensor arus. Semakin besar arus terukur, maka akan semakin besar pula tegangan keluaran sensor. Untuk tegangan asli didapatkan dari pengurangan tegangan *output* dengan 2.5 volt, karena pada sensor arus ini disetting dengan nilai *offset* sebesar 2.5 volt.

Sama halnya dengan data sebelumnya, data perubahan tegangan dan arus terhadap perubahan beban dari data diatas dapat dilihat pada gambar 4.4 dan 4.5 berikut :



**Gambar 4.4** Hasil grafik uji coba tegangan di lampu LHE

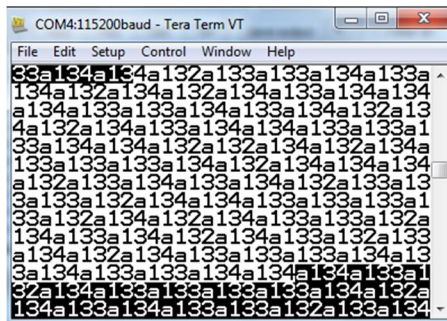


**Gambar 4.5** Hasil grafik uji coba tegangan di lampu LHE

#### 4.4 Pengujian Komunikasi *Serial* ke PC

Komunikasi mikrokontroler dengan PC dilakukan dengan perantara *serial* RS-232. Komunikasi tersebut digunakan untuk mengirim data dari mikrokontroler ke PC untuk keperluan pengolahan data ke aplikasi *Delphi*. Untuk menguji benar tidaknya data yang dikirimkan digunakan aplikasi Tera Term.

Proses pengujian dilakukan dengan membuat program berupa data ADC dan dikirim ke PC dengan media aplikasi Tera Term. Hasil pengiriman data ke PC ditunjukkan pada Gambar 4.6.



**Gambar 4.6** Hasil pengiriman data ke PC dengan serial RS-232

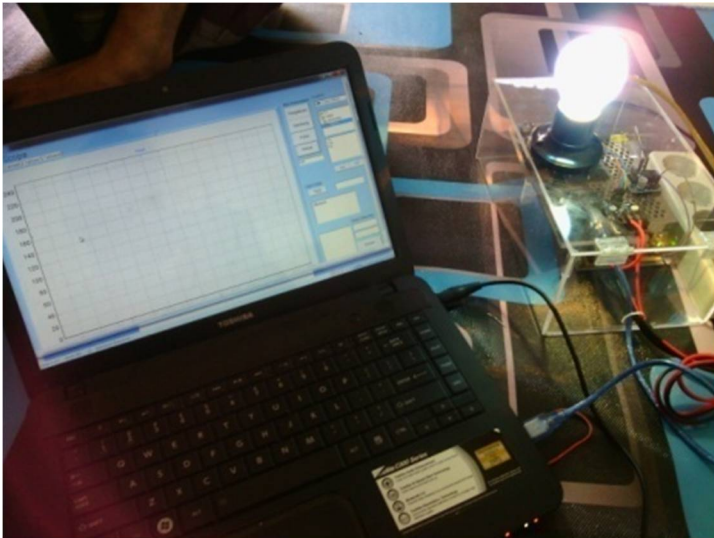
Sesuai yang ditunjukkan pada Gambar 4.6 dapat dilihat bahwa pengiriman data dari mikrokontroler ke PC melalui komunikasi *serial* RS-232 adalah bagus dan bisa dipakai untuk proses pengolahan data dalam sistem.

#### 4.5 Pengujian Alat keseluruhan ke *Delphi*

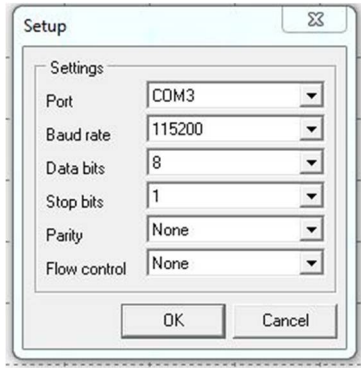
Saat dilakukan pengujian alat secara keseluruhan, dilakukan seluruh koordinasi semua sensor, prototipe jaringan dan tampilan pada PC monitoring.

Tahapan dalam melakukan pengujian *software* monitoring harmonisa arus listrik diperlihatkan dalam urutan berikut ini :

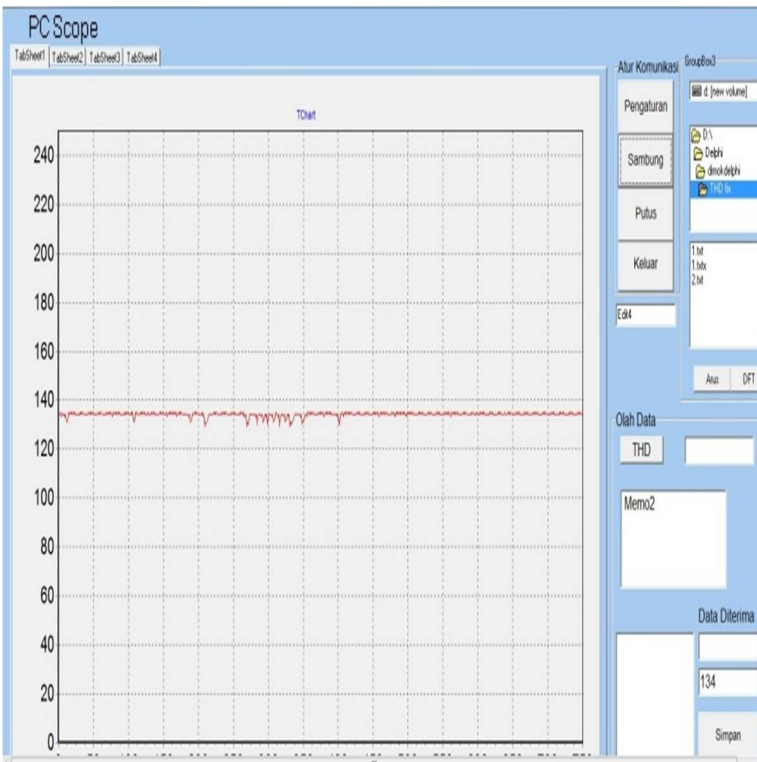
1. Menghubungkan perangkat *hardware* dengan komputer menggunakan kabel *USB to Serial* yang ditunjukkan pada gambar 4.7. Hal ini dilakukan untuk mengetahui apakah antara *hardware* dengan komputer dapat tersambungkan dengan baik. Seperti yang ditunjukkan pada gambar 4.8 yang melakukan setting koneksi dan gambar 4.9 uji coba *delphi* tanpa beban.



**Gambar 4.7** Wiring prototipe dengan PC



**Gambar 4.8** Setting koneksi *delphi* dengan *hardware*

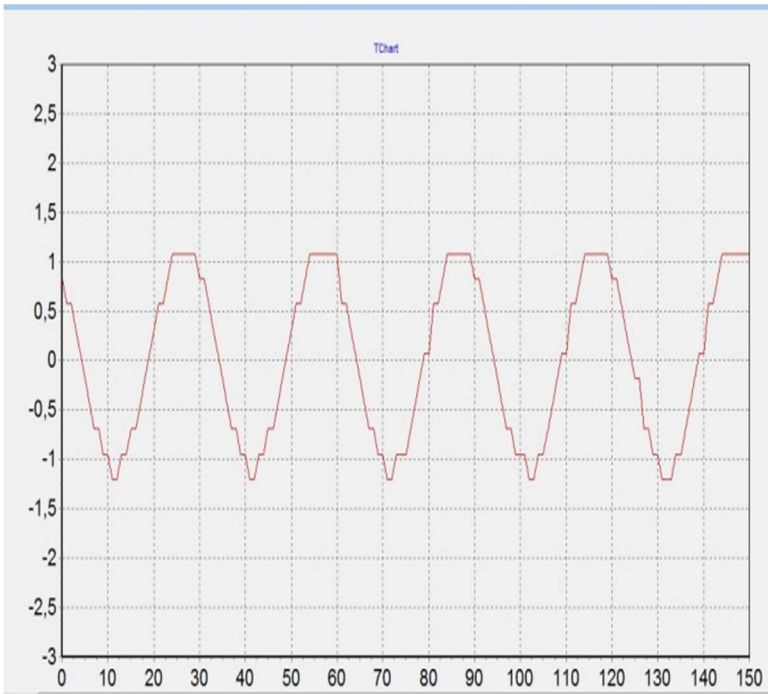


**Gambar 4.9** *Delphi* sudah terhubung dengan *hardware*

Dari gambar 4.9, terlihat bahwa garis yang berwarna merah menandakan antara *hardware* dengan perangkat komputer telah tersambung dengan baik sebagai mana mestinya. Garis merah menandakan kondisi saat belum diberikan pembebanan sehingga bentuknya hanya garis lurus.

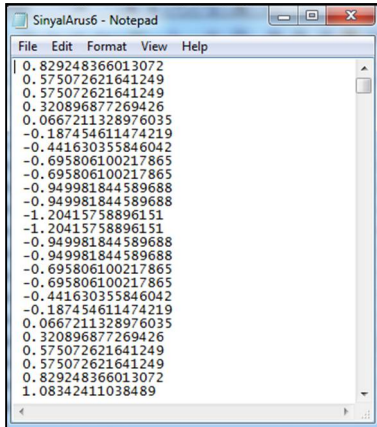
2. Dapat monitoring gelombang arus yang ditunjukkan pada gambar 4.10 dan menyimpan *sampling* arus sebanyak 2500 *sampling* dalam bentuk “.txt” (*notepad*).

*Form* monitoring ini, juga harus dapat menyimpan *sampling* data yang telah masuk. Gambar 4.11 memperlihatkan data arus yang telah tersimpan dalam bentuk “.txt” :



**Gambar 4.10** bentuk gelombang arus pada beban linier diprototipe dari *sampel* 1 – 150





**Gambar 4.11** Hasil data *sampling*

Gambar 4.11 merupakan hasil *sampling* arus sebanyak 2500 *sampling* yang berhasil disimpan dalam bentuk “.txt”. dari data inilah yang nantinya akan digunakan untuk menganalisa apakah beban tersebut berpotensi menyumbangkan gangguan harmonisa atau tidak.

#### 4.5.1 Pengujian Menggunakan Lampu Pijar sebagai Beban

Dalam pengujian ini, dilakukan pengukuran arus menggunakan avometer yang ditampilkan pada *Delphi*. Dalam pengujian ini untuk mengetahui perubahan arus menggunakan perubahan beban lampu pijar yang terpasang. Data perubahan arus dapat dilihat pada tabel 4.5 berikut :

**Tabel 4.5** Data Tampilan Arus pada *Delphi*

No.	Beban (Watt)	Arus Pengukuran (Amper)	Arus Tampilan <i>Delphi</i> (Amper)	% Error
1.	50	0,19	0.185	2,63 %
2.	100	0,4	0.38	5 %
3.	150	0,61	0.58	4,91 %
4.	200	0,82	0.79	3,65 %
				4.047%

Dari data-data diatas yang diperoleh, dapat dicari nilai error dari pengukuran menggunakan Avometer dengan pengukuran menggunakan *Delphi*.

Persentase *error* diperoleh dengan rumus :

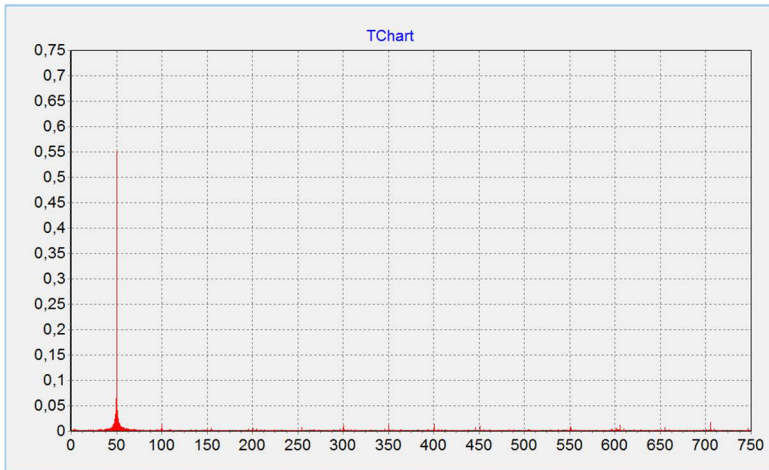
$$\% \text{ Error} = \frac{[\text{Nilai Avometer} - \text{Nilai Delphi}]}{\text{Nilai Avometer}} \times 100\% \dots\dots\dots (4.1)$$

Pada tabel 4.5, nilai yang tercantum pada kolom kesalahan relatif dari hasil pengukuran nilai arus r.m.s pada beban dibandingkan dengan Avometer adalah 4.047%. hal ini berarti prototipe memiliki kinerja yang sangat baik untuk dilakukan monitoring harmonisa.

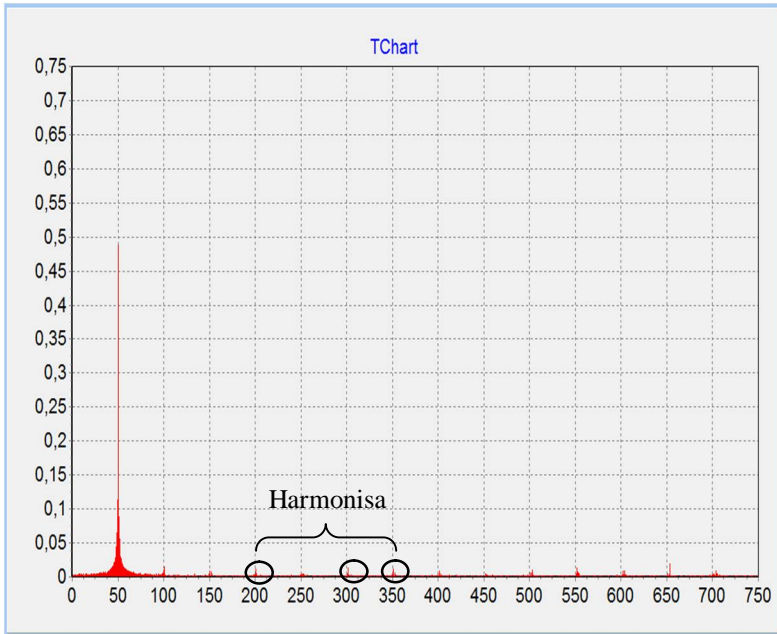
#### 4.5.2 Pengujian Hasil DFT

Pengujian DFT memiliki tujuan untuk mengetahui hasil dari proses DFT dalam menampilkan amplitudo arus pada frekuensi dasar maupun frekuensi kelipatan dari frekuensi dasar untuk menganalisis nilai THDi. Untuk mengetahui hasil dari proses DFT pada prototipe, gelombang arus yang ditampilkan pada borland *delphi* disimpan dengan jumlah data sebanyak 2500 data dalam bentuk file .txt. Setelah itu, data dimasukkan ke borland *delphi* untuk proses DFT.

Pada pengujian DFT ini dilakukan dengan pengukuran pada beban linier dengan daya sebesar 275 Watt. Pada gambar 4.12 menunjukkan hasil dari proses DFT.



**Gambar 4.12** Nilai frekuensi dari frekuensi dasar dari hasil proses pengukuran beban linier



**Gambar 4.13** Harmonisa keluar pada kelipatan 50 Hz

Pada gambar 4.12 dan gambar 4.13 menunjukkan spektrum dari hasil gelombang arus atau amplitudo arus pada domain frekuensi hasil dari proses DFT. Spektrum pada gambar diatas menunjukkan frekuensi dasar 50 HZ mempunyai nilai amplitudo yang lebih tinggi dari pada yang lain dan nilai amplitudo pada frekuensi kelipatan dari frekuensi dasar 50 Hz muncul.

### 4.5.3 Pengujian Level THDi

Pengujian tingkat THDi pada beban dilakukan untuk mengetahui kemampuan dari prototipe untuk mendeteksi dan memonitoring THDi yang merupakan langkah selanjutnya untuk mengidentifikasi jenis peralatan elektronik di Rumah Tangga. Pada pengujian ini, prototipe akan dikenakan beban yang berupa lampu dan peralatan elektronik untuk mengetahui besarnya nilai *Total Harmonic Distorsion* (THD) pada beban tersebut. Tabel 4.6 berikut akan memperlihatkan data hasil pengukuran nilai THD.

**Tabel 4.6** Data perbandingan THDi *Delphi* dengan Fluke

<b>Tes Uji</b>	<b><i>Delphi</i></b>	<b>Fluke</b>	<b>%error</b>
<b>Laptop</b>	35,83%	39,9%	11,35%
<b>LHE</b>	25,8%	29,5%	14%
<b>Lampu Pijar 100 Watt</b>	5,03%	5,1%	0,01%

Dari data-data yang telah didapatkan, dapat dicari % *error* antara pengukuran dengan Fluke dengan arus tampilan *Delphi*. Rumus yang digunakan yaitu :

$$\% \text{ Error} = \frac{[\text{Nilai spesifikasi} - \text{Terukur (manual)}]}{\text{Nilai spesifikasi}} \times 100\%$$

Dari data error diatas, ketepatan alat dengan *software Delphi* dapat dikatakan tepat, karena % *error* yang didapatkan tidak lebih dari 10%. Dari pengujian diatas juga dapat dilihat bahwa beban – beban yang berpotensi menimbulkan gangguan harmonisa arus pada sistem distribusi tenaga listrik adalah beban – beban yang dibuat dengan tujuan penghematan tenaga listrik seperti LHE (Lampu Hemat Energi), laptop, dan yang lainnya. Beban-beban tersebut dikatan berpotensi, karena setelah dilakukan analisa, didapatkan hasil grafik yang menunjukkan terdapat sinyal-sinyal lain yang dapat mengganggu sinyal fundamental dan juga dilihat dari besarnya nilai *Total Harmonic Distorsi* (THD) mencapai lebih dari 10%.

Dengan hasil analisis tersebut dapat dilakukan pengujian berikutnya dengan melakukan tes semua peralatan uji secara bergantian. Tabel 4.7 berikut menampilkan hasil uji THDi peralatan rumah tangga yang berpotensi menghasilkan harmonisa.

**Tabel 4.7** Data hasil uji peralatan rumah tangga

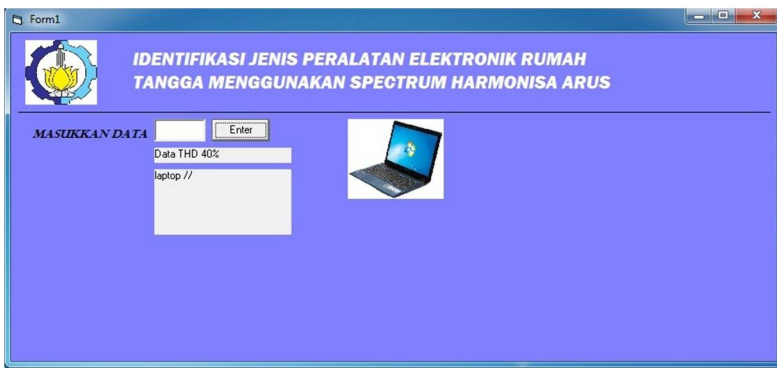
<b>Kipas Angin</b>	<b>Lampu LHE</b>	<b>Laptop</b>	<b>Setrika</b>	<b>THD (1)</b>	<b>THD (2)</b>
1	0	0	0	8,26	9,37
1	1	0	0	8,29	8,55
1	1	1	0	18,07	20,31
1	1	1	1	3,7	4,2

**Tabel 4.7** Data hasil uji peralatan rumah tangga (lanjutan)

Kipas Angin	Lampu LHE	Laptop	Setrika	THD (1)	THD (2)
1	1	0	1	1,71	1,39
1	0	1	0	19,52	29,44
1	0	1	1	2,48	5,94
1	0	0	1	1,2	1,9
0	1	0	0	25,04	25,80
0	1	1	0	24,52	21,67
0	1	1	1	5,68	4,85
0	1	0	1	2,13	1,72
0	0	1	0	43,48	35,83
0	0	1	1	2,93	7,09
0	0	0	1	1,54	140

#### 4.5.4 Identifikasi Peralatan Elektronik Rumah Tangga Menggunakan Visual Basic

Dari hasil THDi pada *Delphi* kemudian dilakukan identifikasi pada program VB 6.0 dengan membuat *range* THDi pada tiap peralatan elektronik rumah tangga. Dengan adanya identifikasi ini dapat diketahui jenis peralatan tiap elektronik yang memiliki nilai THDi yang telah dimasukkan. Hasil Identifikasi dapat dilihat pada Gambar 4.14



**Gambar 4.14** Hasil Identifikasi Menggunakan *Visual Basic*

Dari gambar 4.14 dapat dilihat bahwa setelah dimasukkan suatu nilai dapat diketahui nilai yang telah dimasukkan tersebut termasuk dalam *range* nilai peralatan elektronik atau tidak ada peralatan elektronik. Apabila nilai yang dimasukkan masuk dalam *range* nilai laptop maka akan diketahui nilai tersebut termasuk dalam *range* THDi laptop. Sedangkan apabila dimasukkan nilai yang tidak masuk pada setting *range* nilai THDi peralatan elektronik maka akan muncul Tidak Ada Peralatan.

*Halaman Ini Sengaja Dikosongkan*

## **BAB V**

### **PENUTUP**

Dari seluruh kegiatan Tugas Akhir ini mulai dari perancangan dan pembuatan alat sampai pengujian dan analisa, maka dapat ditarik kesimpulan dan saran sebagai berikut.

#### **5.1 Kesimpulan**

Dari Tugas Akhir yang telah dibuat, dapat ditarik beberapa kesimpulan dari hasil data yang telah didapatkan, yaitu :

1. Prototipe layak digunakan karena *%error* antara alat pengukuran dan prototipe rata-rata *%error*-nya di bawah 10% sesuai Tabel 4.5.
2. Dari data dapat dilihat bahwa setrika mampu mengurangi THDi pada beban non-linier sesuai hasil Tabel 4.7.
3. Dari hasil pengambilan data saat menggunakan beban-beban yang berbeda dapat diketahui bahwa lampu pijar tidak berpotensi menyumbangkan gangguan harmonisa arus, namun lampu hemat energi memiliki potensi menyumbangkan gangguan harmonisa arus karena dilihat pada grafik beban tersebut memiliki sinyal pada orde ganjil dan juga nilai THDi yang tinggi.
4. Prototipe yang dibuat dalam tugas akhir ini mampu memantau keberadaan harmonisa arus hingga orde 15 (frekuensi 750 Hz).

#### **5.2 Saran**

Dalam setiap pembuatan alat pasti jauh dari kata sempurna, kekurangan dan kesalah masih banyak ditemukan dalam alat ini. Saran untuk menyempurnakan alat ini, agar lebih sempurna lagi yaitu :

1. Untuk meningkatkan keandalan prototipe perlu dilakukan pengintegrasian prototipe dengan sistem web server sehingga data *online* dapat dikirim dan ditampilkan melalui media internet.
2. Dalam proses ADC memerlukan jenis mikrokontroler yang lebih baik seperti ARM.
3. Agar data sensor arus menghasilkan error yang minimal, setting sensor arus harus dilakukan lebih teliti atau menggunakan sensor arus yang lebih teliti lagi.



*Halaman Ini Sengaja Dikosongkan*

## DAFTAR PUSTAKA

- [1] **Anggriawan, D.O**, *RANCANG BANGUN SISTEM PENDETEKSIAN DAN MONITORING HARMONISA MENGGUNAKAN METODE DFT*, Tugas Akhir Elektro ITS, 2013
- [2] **Setiawan, A**, “*KAJIAN PENGARUH HARMONISA TERHADAP SISTEM TENAGA LISTRIK*”, **Jurnal Eltek, Vol. 05, No. 02, Oktober, 2007**
- [3] **Arrillaga, J., Watson, N.R.**, “*POWER SYSTEM HARMONICS SECOND EDITION*”, John Wiley & Sons Ltd, England, Ch. 1,2,5, 2003
- [4] **Andrianto dan Heri**, “*PEMROGRAMAN MIKROKONTROLLER AVR ATMEGA 16 MENGGUNAKAN BAHASA C (CODEVISION AVR)*”, Informatika, Bandung, 2008
- [5] **Rohman .Y, Budikarso .A, Amran .H** , “*RANCANG BANGUN SISTEM PENGUKURAN ARUS BERBASIS MIKROKONTROLLER ATMEGA 8535*” Paper PENS, Surabaya
- [6] **Arveliansyah, D., & Priyono, A. (t.thn.)**. “*LAPORAN TUGAS BESAR PERANCANGAN MIKROELEKTRONIKAFFT 256 TITIK RADIX 8*”. Dipetik Maret 18, 2015, dari [http://www.academia.edu/4522075/LAPORAN\\_TUGAS\\_BESAR](http://www.academia.edu/4522075/LAPORAN_TUGAS_BESAR)
- [7] **Abdul Kadir**, “*PEMROGRAMAN DATABASE DENGAN DELPHI 7*”, Andi, Yogyakarta, 2004

*Halaman ini sengaja dikosongkan*

## LAMPIRAN 1 LISTING PROGRAM

### Listing Program Code Vision AVR

/\*\*

This program was produced by the  
CodeWizardAVR V2.05.3 Standard  
Automatic Program Generator  
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.  
<http://www.hpinfotech.com>

Project :  
Version :  
Date : 11/05/2013  
Author : Duo KH  
Company : Kiru Company!  
Comments:

Chip type : ATmega16  
Program type : Application  
AVR Core Clock frequency: 16,000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 256

\*/

```
#include <mega16.h>
```

```
#include <delay.h>
```

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

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

```
unsigned char baca,ratusan,puluhan,satuan;
```

```
#define ADC_VREF_TYPE 0x60
```

```
// Read the 8 most significant bits
```

```
// of the AD conversion result
```

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

```

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

// Declare your global variables here

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

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

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

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

```

```

DDRC=0x00;

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

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

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

```

```

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

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

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

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 115200
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x08;

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

// ADC initialization
// ADC Clock frequency: 1000,000 kHz

```

```

// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: ADC Stopped
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

while (1)
{
    // Place your code here
    baca = read_adc(0);
    ratusan = baca/100;
    puluhan = (baca - (ratusan*100))/10;
    satuan = baca - (ratusan*100) - (puluhan*10);
    printf("a%d%d%d",ratusan,puluhan,satuan);
    delay_us(400);
}
}

```

### **Listing Program Harmonisa**

```

unit Unit1;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms,
    Dialogs, ExtCtrls, TeeProcs, TeEngine, Chart, StdCtrls, Series,
    ComCtrls, CPort, Buttons, FileCtrl;

type
    TForm1 = class(TForm)
        ComPort1: TComPort;

```



ComDataPacket1: TComDataPacket;  
Button1: TButton;  
Button2: TButton;  
Button3: TButton;  
Button4: TButton;  
GroupBox1: TGroupBox;  
Timer1: TTimer;  
OpenDialog1: TOpenDialog;  
SaveDialog1: TSaveDialog;  
Button6: TButton;  
Memo1: TMemo;  
Label1: TLabel;  
GroupBox2: TGroupBox;  
Edit1: TEdit;  
Label2: TLabel;  
Edit2: TEdit;  
PageControl1: TPageControl;  
TabSheet1: TTabSheet;  
TabSheet2: TTabSheet;  
Chart1: TChart;  
Series1: TLineSeries;  
Chart2: TChart;  
Series2: TLineSeries;  
TabSheet3: TTabSheet;  
Chart3: TChart;  
Series3: TLineSeries;  
GroupBox3: TGroupBox;  
FileListBox2: TFileListBox;  
DirectoryListBox1: TDirectoryListBox;  
DriveComboBox1: TDriveComboBox;  
Button5: TButton;  
TabSheet4: TTabSheet;  
Chart4: TChart;  
Series4: TBarSeries;  
Button7: TButton;  
Edit3: TEdit;  
Button8: TButton;  
Memo2: TMemo;  
Edit4: TEdit;  
procedure Button1Click(Sender: TObject);  
procedure Button3Click(Sender: TObject);

```

procedure Button2Click(Sender: TObject);
procedure packet(Sender: TObject; const Str: String);
procedure FormCreate(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure delay(lama:real);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure dft();
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  y,speed,mulai_simpan : integer;
  a : string;
  b : string;
  c : Longint;
  terima : boolean;
  counter,tipe,dataint_acc,dataint_gyro,dataint_mag: longint;
  ref,awal,akhir,lama:longint;
  x,Xr, Xi, Mx : array[0..3000] of real;
  buf : array[0..15] of real;
implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var start : integer;
begin
    y := 0 ;
    Series1.Clear;
    terima := false;
    comport1.Open;
end;

```

```

function TextToHex ( const s: string ): string;
var
  x: integer;
begin
  for x := 1 to Length(s) do
    result := result + IntToHex(Integer(s[x]),2);
  end;

```

```

function HexToBin(Hexadecimal: string): string;
const
  BCD: array [0..15] of string =
    ('0000', '0001', '0010', '0011', '0100', '0101', '0110', '0111',
     '1000', '1001', '1010', '1011', '1100', '1101', '1110', '1111');
var
  i: integer;
begin
  for i := Length(Hexadecimal) downto 1 do
    Result := BCD[StrToInt('$' + Hexadecimal[i])] + Result;
  end;

```

```

function BinToInt(Value: String): LongInt;
var i: Integer;
begin
  Result:=0;
  //remove leading zeroes
  while Copy(Value,1,1)='0' do
    Value:=Copy(Value,2,Length(Value)-1) ;
  //do the conversion
  for i:=Length(Value) downto 1 do
    if Copy(Value,i,1)='1' then
      Result:=Result+(1 shl (Length(Value)-i)) ;
  end;

```

```

function HexToInt_a(HexNum1:char ; HexNum2:char ): LongInt;
begin
  Result:=StrToInt('$' + HexNum1 + HexNum2) ;
end;

```

```

procedure TForm1.delay(lama:real); //prosedur delay untuk grafik
var ref:longint;

```

```

begin
  ref:=gettickcount;
  repeat application.processmessages;
  until ((gettickcount-ref)>=lama);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  comport1.ShowSetupDialog;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  comport1.Close;
end;

procedure TForm1.packet(Sender: TObject; const Str: String);
var
  nilai : array [1..3] of integer;
  out,binbin,binbin2,binbin4,binbin6: string;
  angka1,angka2,angka3,angka10 : longint;
  arus,tegangan : real;

begin
  //out := TextToHex(Str);
  Edit1.Text := Str;
  out := TextToHex(Str);
  binbin := HexToBin(out[1]);binbin2 := HexToBin(out[2]);binbin4 :=
    HexToBin(out[4]);
  binbin6 := HexToBin(out[6]);//binbin8 := HexToBin(out[8]);
  angka1 := BintToInt(binbin2); angka2 := BintToInt(binbin4); angka3 :=
    BintToInt(binbin6);
  angka10 := angka1*100 + angka2*10 + angka3;
  arus :=0.105988*(angka10-127.5);
  //edit6.Text := inttostr(angka10);
  series1.AddXY(y,angka10);
  series2.AddXY (y,arus);
  y:= y+1;
  if (mulai_simpan = 1) then

```

```

begin
    memo1.Lines.Add(' '+floatToStr(arus)+' ');
    //ref:=gettickcount;
end;
if (y=2500) then
begin
akhir := gettickcount;
lama := akhir - awal;
if (mulai_simpan = 1) then
begin
if savedialog1.execute then
    memo1.Lines.SavetoFile(savedialog1.FileName);
edit2.Text := inttostr(lama);
mulai_simpan := 0;
end;
y := 0;
series1.Clear;
Series2.Clear;
end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
filelistbox2.Mask := '*.txt';
directorylistbox1.FileList := filelistbox2;
drivecombobox1.DirList := directorylistbox1;

//edit5.Text := inttostr(Trackbar1.Position)+'%';
memo1.clear();
end;

procedure TForm1.TrackBar1Change(Sender: TObject);
begin
//speed := Trackbar1.Position;
Comport1.Write(speed,1);
//edit5.Text := inttostr(Trackbar1.Position)+'%';
end;

procedure TForm1.Button5Click(Sender: TObject);
var
j,i,k : integer;

```

```

imax, irms : real;
myFile: TextFile;

begin
assignFile(myFile,filelistbox2.FileName);
Reset(myfile);
i:= 0;
while not Eof(myFile) do
begin
  Read(myFile, x[i]);
  i:=i+1;
end;
CloseFile(myFile);
imax:=0;
series3.Clear;
for i:= 0 to 2499 do
begin
series3.AddXY(i,x[i]);
// DFT
  if ((x[i])>=(imax)) then imax:=x[i];
end;

irms:=imax/sqrt(2);
Edit4.Text:=floattostr(irms);

end;

procedure TForm1.Button6Click(Sender: TObject);
begin
//if savedialog1.execute then
  memo1.Lines.SavetoFile(savedialog1.FileName);
mulai_simpan := 1;
y:= 0;
awal := gettickcount;
dft();
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
close;

```

```

end;

procedure TForm1.Button4Click(Sender: TObject);
begin
close;
end;

begin

procedure TForm1.Button7Click(Sender: TObject);
var ax, bx, delta : real;
j,k,n, blok : integer;
begin
delta := 1/(2500*0.000664011);

for k:=0 to 1249 do
begin
ax:=0;bx:=0;
for j:=0 to 2499 do
begin
Xr[k]:=ax+x[j]*cos(2*3.14*k*j/2500);
Xi[k]:=bx-x[j]*sin(2*3.14*k*j/2500);
ax:=Xr[k]; bx:=Xi[k];
end;
Mx[k]:=sqrt(sqr(Xr[k])+sqr(Xi[k]))/2500;
Series4.AddXY(k*delta, Mx[k]);
end;
{ for blok:=1 to 15 do
begin
for n := (82*blok)-4 to (82*blok)+6 do
begin
buf[n] := Mx[82*blok-4];
end;
end;
for blok:=1 to 15 do
begin
for n := (82*blok)-4 to (82*blok)+6 do
begin
buf[n] := Mx[82*blok-4];
end;
end;
end;

```

```

    }
end;
procedure TForm1.Button8Click(Sender: TObject);
var
abc,scan_freq: integer;
delta,nilai_max,bag_atas,nilai_THD : real;
begin

delta := 1/(2500*0.000664011);
bag_atas:=0;
for abc:=1 to 6 do
begin
nilai_max:=0;
for scan_freq:= round(((2*abc)*50/delta)) to
round(((2*abc)*50/delta)) do
begin
if (Mx[scan_freq]>nilai_max) then
begin
nilai_max:=Mx[scan_freq];
end;
end;
bag_atas:=bag_atas+sqr(nilai_max);
end;
nilai_max:=0;
for scan_freq:= round((50/delta)) to round((50/delta)) do
begin
if (Mx[scan_freq]>nilai_max) then
begin
nilai_max:=Mx[scan_freq];
end;
end;
nilai_THD:=sqrt(bag_atas)/nilai_max*100;
edit3.Text:=floattostr(nilai_THD);
end;
end.

```

### **Listing Program LOGIN Identifikasi di Visual Basic 6.0**

```

Private Sub Command1_Click()
If Text1.Text = "a" And Text2.Text = "a" Then
Unload Me

```



```
Form1.Visible = True
End If
End Sub
```

### **Listing Program Identifikasi di Visual Basic 6.0**

```
Private Sub Command1_Click()

Label2.Caption = ""
Label3.Caption = ""
Image1.Visible = False
Image2.Visible = False
Image3.Visible = False
Image4.Visible = False
Image5.Visible = False
Image6.Visible = False
Image7.Visible = False
Image8.Visible = False
Image9.Visible = False
Image10.Visible = False
Image11.Visible = False
Image12.Visible = False
Image13.Visible = False
Image14.Visible = False
Image15.Visible = False
Image16.Visible = False

Label2.Caption = "Data THD " & Text1.Text & "%"
If Val(Text1.Text) > 8.26 And Val(Text1.Text) < 9.37 Then
Label3.Caption = Label3.Caption & "Kipas // "
Image1.Visible = True
End If

If Val(Text1.Text) >= 8.29 And Val(Text1.Text) < 8.55 Then
Label3.Caption = Label3.Caption & "Kipas dan Lampu // "
Image16.Visible = True
End If

If Val(Text1.Text) >= 18.07 And Val(Text1.Text) < 20.31 Then
Label3.Caption = Label3.Caption & "Kipas & Lampu & Laptop // "
Image15.Visible = True
```

End If

```
If Val(Text1.Text) >= 3.7 And Val(Text1.Text) < 4.2 Then
Label3.Caption = Label3.Caption & "Kipas, Lampu, laptop, dan setrika
// "
Image4.Visible = True
End If
```

```
If Val(Text1.Text) >= 1.39 And Val(Text1.Text) < 1.71 Then
Label3.Caption = Label3.Caption & "Kipas, Lampu, dan setrika // "
Image11.Visible = True
End If
```

```
If Val(Text1.Text) >= 19.52 And Val(Text1.Text) < 29.44 Then
Label3.Caption = Label3.Caption & "Kipas dan laptop // "
Image2.Visible = True
End If
```

```
If Val(Text1.Text) >= 2.48 And Val(Text1.Text) < 5.94 Then
Label3.Caption = Label3.Caption & "Kipas, laptop, dan setrika // "
Image13.Visible = True
End If
```

```
If Val(Text1.Text) >= 1.2 And Val(Text1.Text) < 1.9 Then
Label3.Caption = Label3.Caption & "Kipas dan setrika // "
Image9.Visible = True
End If
```

```
If Val(Text1.Text) >= 25.04 And Val(Text1.Text) < 25.8 Then
Label3.Caption = Label3.Caption & "Lampu // "
Image3.Visible = True
End If
```

```
If Val(Text1.Text) >= 21.67 And Val(Text1.Text) < 24.52 Then
Label3.Caption = Label3.Caption & "Lampu dan laptop // "
Image10.Visible = True
End If
```

```
If Val(Text1.Text) >= 4.85 And Val(Text1.Text) < 5.68 Then
Label3.Caption = Label3.Caption & "Lampu, laptop, dan setrika // "
Image14.Visible = True
```

End If

```
If Val(Text1.Text) >= 1.72 And Val(Text1.Text) < 2.13 Then
Label3.Caption = Label3.Caption & "Lampu dan setrika // "
Image8.Visible = True
End If
```

```
If Val(Text1.Text) >= 35.83 And Val(Text1.Text) < 43.83 Then
Label3.Caption = Label3.Caption & "laptop // "
Image6.Visible = True
End If
```

```
If Val(Text1.Text) >= 2.93 And Val(Text1.Text) < 7.09 Then
Label3.Caption = Label3.Caption & " laptop dan setrika // "
Image12.Visible = True
End If
```

```
If Val(Text1.Text) >= 1.4 And Val(Text1.Text) < 1.54 Then
Label3.Caption = Label3.Caption & " setrika // "
Image7.Visible = True
End If
```

```
If Label3.Caption = "" Then
Label3.Caption = "Tidak ada peralatan"
End If
```

```
Text1.Text = ""
End Sub
```

```
Private Sub Form_Load()
Form2.Show
Form1.Hide
```

```
Image1.Visible = False
Image2.Visible = False
Image3.Visible = False
Image4.Visible = False
Image5.Visible = False
Image6.Visible = False
Image7.Visible = False
Image8.Visible = False
```

```
Image9.Visible = False  
Image10.Visible = False  
Image11.Visible = False  
Image12.Visible = False  
Image13.Visible = False  
Image14.Visible = False  
Image15.Visible = False  
Image16.Visible = False  
End Sub
```

## LAMPIRAN II

### LISTING DATASHEET

#### 1. Datasheet ATmega16

##### Features

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



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

ATmega16  
ATmega16L

Preliminary

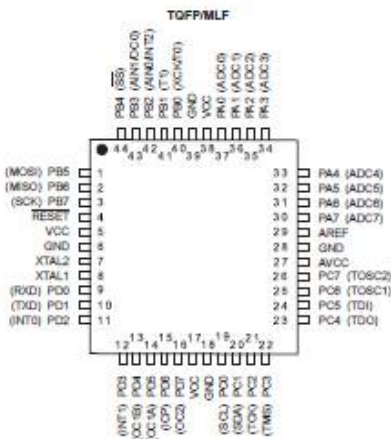
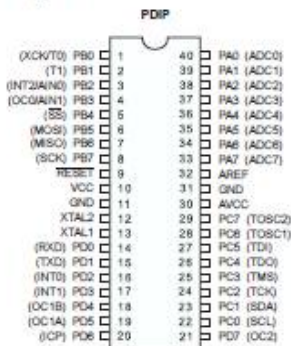
Rev. 2490E-AVR-10b2





## Pin Configurations

Figure 1. Pinouts ATmega16



## Disclaimer

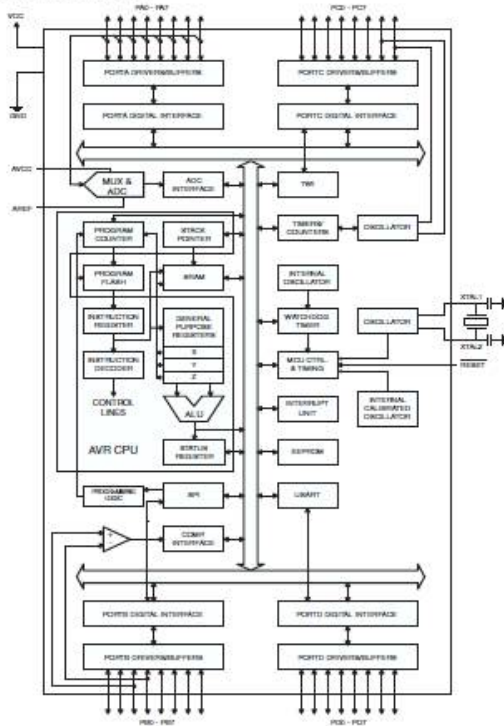
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## Overview

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

## Block Diagram

Figure 2. Block Diagram





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

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

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

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

#### Pin Descriptions

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



<b>Port B (PB7..PB0)</b>	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega16 as listed on page 55.</p>
<b>Port C (PC7..PC0)</b>	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 58.</p>
<b>Port D (PD7..PD0)</b>	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega16 as listed on page 60.</p>
<b>RESET</b>	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 35. Shorter pulses are not guaranteed to generate a reset.</p>
<b>XTAL1</b>	<p>Input to the Inverting Oscillator amplifier and input to the internal clock operating circuit.</p>
<b>XTAL2</b>	<p>Output from the Inverting Oscillator amplifier.</p>
<b>AVCC</b>	<p>AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to <math>V_{CC}</math>, even if the ADC is not used. If the ADC is used, it should be connected to <math>V_{CC}</math> through a low-pass filter.</p>
<b>AREF</b>	<p>AREF is the analog reference pin for the A/D Converter.</p>
<b>About Code Examples</b>	<p>This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.</p>

## 2. Dataheet Sensor ACS712

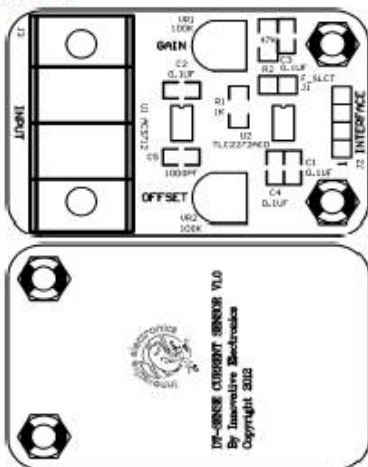
### DT-Sense Current Sensor

**DT-Sense Current Sensor** merupakan suatu modul sensor arus yang menggunakan IC sensor arus linier berbasis Hall-Effect ACS712 produksi Allegro. Sensor arus ini dapat digunakan untuk mengukur arus AC atau DC. Untuk modul DT-SENSE dengan tipe with OpAmp, telah ditambahkan rangkaian OpAmp sehingga sensitivitas pengukuran arus dapat lebih ditingkatkan dan dapat mengukur perubahan arus yang lebih kecil. Sensor ini digunakan pada aplikasi-aplikasi di bidang industri, komersial, maupun komunikasi. Contoh aplikasinya antara lain untuk sensor kontrol motor, deteksi dan manajemen penggunaan daya, sensor untuk switch-mode power supply, sensor proteksi terhadap overcurrent, dan lain sebagainya.

#### Spesifikasi

- Barbasis ACS712 dengan fitur:
  - Rise time output = 5  $\mu$ s.
  - Bandwidth sampai dengan 80 kHz.
  - Total kesalahan output 1,5% pada suhu kerja  $T_A = 25^\circ\text{C}$ .
  - Tahanan konduktor internal 1,2 m $\Omega$ .
  - Tegangan isolasi minimum 2,1 kV<sub>RMS</sub> antara pin 1-4 dan pin 5-8.
  - Sensitivitas output 185 mV/A.
  - Mampu mengukur arus AC atau DC hingga 5 A.
  - Tegangan output proporsional terhadap input arus AC atau DC.
- Tegangan kerja 5 VDC.
- Dilengkapi dengan OpAmp untuk menambah sensitivitas output (untuk tipe With OpAmp).

#### Tata Letak



Alokasi Pin INTERFACE (J2)		
Pin	Nama	Fungsi
1	VCC	Input Tegangan 5 VDC
2	Out	Output Output dari sensor
3	Out_Amp	Output dari OpAmp
4	Ground	- Titik referensi ground

Tegangan output ACS712 terhubung ke pin Out dan tegangan output rangkaian OpAmp terhubung ke Out\_Amp.

Pada tipe **Without OpAmp**, pin Out\_Amp dapat diabaikan.

Rumus tegangan pada pin Out =  $2,5 \pm [0,185 \times I]$  Volt Dimana I = arus yang terdeteksi dalam satuan Ampere.

Pada tipe **With OpAmp**, modul sudah dilengkapi dengan rangkaian OpAmp yang dapat digunakan untuk meningkatkan sensitivitas dan mengubah offset pada tegangan output OpAmp (pin Out\_Amp).

Sensitivitas atau Gain diatur melalui VR1 sedangkan offset diatur melalui VR2.

Pada tipe **With OpAmp** telah dilengkapi pula dengan filter untuk mengurangi noise dengan efek samping bandwidth output menjadi lebih kecil.

Bandwidth output dapat dibuat kembali maksimal (menghilangkan filter) dengan melapas jumper F\_SLECT [J1].

#### Isi CD/DVD

- Manual DT-Sense Current Sensor.
- Data sheet.
- Website Offline Innovative Electronics.

#### Prosedur Pengujian

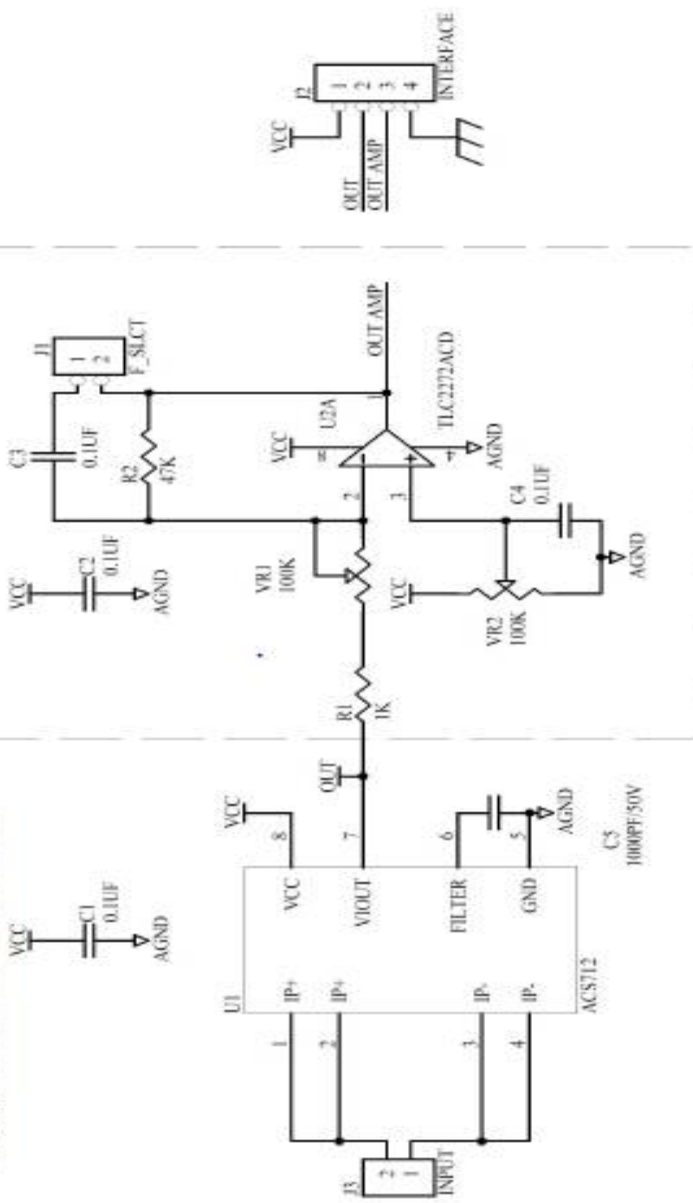
Pengujian sederhana dapat dilakukan dengan langkah-langkahnya sebagai berikut:

- Hubungkan VCC (pin 1) pada modul DT-SENSE dengan sumber tegangan 5 Volt dan pin 4 dengan ground.
- Ukur tegangan Out (pin 2) pada modul DT-SENSE dengan voltmeter.
- Jika tidak ada arus yang lewat melalui input, maka Out akan bernilai sekitar 2,5 Volt.
- Jika modul DT-SENSE CURRENT SENSOR dilengkapi dengan OpAmp, maka atur Offset (VR2) sehingga nilai Out\_Amp menjadi sekitar 2,5 Volt.
- Lewatkan arus listrik melalui jalur input. Semakin besar arus yang melawati input, semakin besar pula perubahan tegangan pada pin Out dan pin Out\_Amp.
- Karena sifat dari rangkaian OpAmp yang digunakan, jika tegangan Out lebih kecil dari 2,5 Volt, maka tegangan Out\_Amp akan lebih besar dari pada 2,5 Volt.
- Sebaliknya, jika tegangan Out lebih besar dari 2,5 Volt, maka tegangan Out\_Amp akan lebih kecil dari pada 2,5 Volt.

\* Terima Kasih atas kepercayaan Anda menggunakan produk kami, bila ada keawalan, pertanyaan, atau saran mengenai produk ini silakan menghubungi technical support kami:

support@innovativeelectronics.com

Copyright © 2012 Innovative Electronics



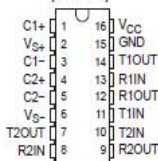
### 3. Datasheet MAX-232

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

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

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

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



#### description/ordering information

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

#### ORDERING INFORMATION

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

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/ps/package](http://www.ti.com/ps/package).



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

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Product fit, performance, data and warranty may be subject to change without notice.



POST OFFICE BOX 655503 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated

1

**MAX232, MAX232I**  
**DUAL EIA-232 DRIVERS/RECEIVERS**

SLL8047L - FEBRUARY 1989 - REVISED MARCH 2004

**Function Tables**

**EACH DRIVER**

INPUT TIN	OUTPUT TOUT
L	H
H	L

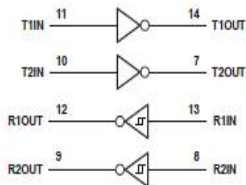
H = high level, L = low level

**EACH RECEIVER**

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



## MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

### absolute maximum ratings over operating free-air temperature range (unless otherwise noted)<sup>†</sup>

Input supply voltage range, $V_{CC}$ (see Note 1)	-0.3 V to 6 V
Positive output supply voltage range, $V_{S+}$	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, $V_{S-}$	-0.3 V to -15 V
Input voltage range, $V_i$ : Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	$\pm 30$ 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, $\theta_{JA}$ (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

<sup>†</sup> 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  $T_J(\max)$ ,  $\theta_{JA}$ , and  $T_A$ . The maximum allowable power dissipation at any allowable ambient temperature is  $P_D = (T_J(\max) - T_A)/\theta_{JA}$ . Operating at the absolute maximum  $T_J$  of 150°C can affect reliability.

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

### recommended operating conditions

		MIN	NOM	MAX	UNIT
$V_{CC}$	Supply voltage	4.5	5	5.5	V
$V_{IH}$	High-level input voltage (T1IN, T2IN)	2			V
$V_{IL}$	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			$\pm 30$	V
$T_A$	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	-40	85	

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

PARAMETER	TEST CONDITIONS	MIN	TYP <sup>‡</sup>	MAX	UNIT
$I_{CC}$	Supply current	$V_{CC} = 5.5$ V, $T_A = 25^\circ$ C		8	10 mA

<sup>‡</sup> All typical values are at  $V_{CC} = 5$  V and  $T_A = 25^\circ$  C.

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



POST OFFICE BOX 988303 • DALLAS, TEXAS 75298

3



**MAX232, MAX232I**  
**DUAL EIA-232 DRIVERS/RECEIVERS**

SLL5047L - 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 <sup>†</sup>	MAX	UNIT
V <sub>OH</sub> High-level output voltage	T1OUT, T2OUT R <sub>L</sub> = 3 kΩ to GND	5	7		V
V <sub>OL</sub> Low-level output voltage <sup>‡</sup>	T1OUT, T2OUT R <sub>L</sub> = 3 kΩ to GND	-7	-5		V
r <sub>O</sub> Output resistance	T1OUT, T2OUT V <sub>OH</sub> = V <sub>OL</sub> = 0, V <sub>O</sub> = ±2 V	300			Ω
I <sub>OCS</sub> <sup>§</sup> Short-circuit output current	T1OUT, T2OUT V <sub>CC</sub> = 5.5 V, V <sub>O</sub> = 0		±10		mA
I <sub>IC</sub> Short-circuit input current	T1IN, T2IN V <sub>I</sub> = 0		200		μA

<sup>†</sup> All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

<sup>‡</sup> 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.

<sup>§</sup> Not more than one output should be shorted at a time.

NOTE 4: Test conditions are C1-C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR Driver slew rate	R <sub>L</sub> = 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		kbit/s

NOTE 4: Test conditions are C1-C4 = 1 μF at V<sub>CC</sub> = 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 <sup>†</sup>	MAX	UNIT
V <sub>OH</sub> High-level output voltage	R1OUT, R2OUT I <sub>OH</sub> = -1 mA	3.5			V
V <sub>OL</sub> Low-level output voltage <sup>‡</sup>	R1OUT, R2OUT I <sub>OL</sub> = 3.2 mA		0.4		V
V <sub>IT+</sub> Receiver positive-going input threshold voltage	R1IN, R2IN V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	1.7	2.4		V
V <sub>IT-</sub> Receiver negative-going input threshold voltage	R1IN, R2IN V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	0.8	1.2		V
V <sub>HY</sub> Input hysteresis voltage	R1IN, R2IN V <sub>CC</sub> = 5 V	0.2	0.5	1	V
r <sub>I</sub> Receiver input resistance	R1IN, R2IN V <sub>CC</sub> = 5, T <sub>A</sub> = 25°C	3	5	7	kΩ

<sup>†</sup> All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

<sup>‡</sup> 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<sub>CC</sub> = 5 V ± 0.5 V.

switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C (see Note 4 and Figure 1)

PARAMETER	TYP	UNIT
t <sub>PLH(R)</sub> Receiver propagation delay time, low- to high-level output	500	ns
t <sub>PCHL(R)</sub> Receiver propagation delay time, high- to low-level output	500	ns

NOTE 4: Test conditions are C1-C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

*Halaman Ini Sengaja Dikosongkan*



## DAFTAR RIWAYAT HIDUP



Nama : Ahmad Naufa Syauqi  
TTL : Bojonegoro, 17 Mei 1994  
Jenis Kelamin : Laki – laki  
Agama : Islam  
Alamat Rumah : Ds. Ngumoakdalem 15/04,  
Kec. Dander, Kab.  
Bojonegoro  
E-mail : ahmadnaufa.s@gmail.com  
Hobi : Berolahraga

### RIWAYAT PENDIDIKAN

2000 – 2006 : MI Negeri 1 Bojonegoro  
2006 – 2009 : MTs Negeri 1 Bojonegoro  
2009 – 2012 : SMA Darul ‘Ulum 1 Unggulan BPP-T  
Jombang  
2012 – sekarang : Bidang Studi Teknik Listrik, Program Studi  
D3 Teknik Elektro, ITS

### PENGALAMAN ORGANISASI DAN KEPANITIAAN

1. Sekertaris umum LDJ Salman Al Farisi D3 Teknik Elektro  
2013/2014
2. Panitia IARC D3 Teknik Elektro 2013/2014

*Halaman ini sengaja dikosongkan*

## DAFTAR RIWAYAT HIDUP



Nama : Abdul Hadi  
TTL : Bangkalan, 15 Juli 1994  
Jenis Kelamin : Laki – laki  
Agama : Islam  
Alamat Rumah : Perum Bangkalan Indah ME  
19 A Bangkalan  
E-mail : abdulhadi303@gmail.com  
Hobi : Membaca

### RIWAYAT PENDIDIKAN

2000 – 2006 : SDN Pejagan 6 Bangkalan  
2006 – 2009 : SMPN 1 Bangkalan  
2009 – 2012 : SMAN 3 Bangkalan  
2012 – sekarang : Bidang Studi Teknik Listrik, Program Studi  
D3 Teknik Elektro, ITS

### PENGALAMAN ORGANISASI DAN KEPANITIAAN

1. Staff LDJ Salman Al Farisi D3 Teknik Elektro 2013/2014
2. Bendahara LDJ Salman Al Farisi D3 Teknik Elektro 2014/2015
3. Panitia IARC D3 Teknik Elektro 2013/2014
4. Asdir Kopma Dr Angka ITS
5. Pembina Lingkungan Hidup di Persada Raya Bangkalan

*Halaman ini sengaja dikosongkan*