

**SISTEM PEMANTAUAN GENERATOR BERBASIS *ARDUINO*
DAN *VISUAL BASIC***

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

ADAM ZULHAN

NIRM : 0031501

AJI KUSUMA

NIRM : 0031502

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**

2018

LEMBAR PENGESAHAN

**SISTEM PEMANTAUAN GENERATOR BERBASIS *ARDUINO* DAN
*VISUAL BASIC***

Oleh :

Adam Zulhan

NIRM : 0031501

Aji Kusuma

NIRM : 0031502


Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,


Pembimbing 1


Yudhi, M.T.

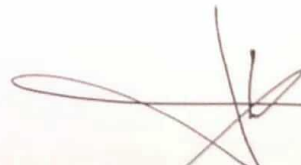
Pembimbing 2


Indra Dwisaputra, M.T.

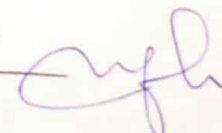
Penguji 1


M. Iqbal Nugraha, M.Eng.

Penguji 2


Surojo, M.T.

Penguji 3


Dr. Parulian Silalahi, M.Pd.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Adam Zulhan NIRM : 0031501

Nama Mahasiswa 2 : Aji Kusuma NIRM : 0031502

Dengan Judul : SISTEM PEMANTAUAN GENERATOR BERBASIS
ARDUINO DAN VISUAL BASIC


Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia diberikan sanksi yang berlaku.

Sungailiat, 29 Juni 2018

Nama Mahasiswa

Tanda Tangan

1. Adam Zulhan


.....

2. Aji Kusuma


.....

ABSTRAK

Pembangkit listrik adalah bagian dari alat industri yang dipakai untuk membangkitkan tenaga listrik dari berbagai sumber tenaga, seperti PLTU, PLTD, PLTN, dan PLTA. Bagian utama dari pembangkit listrik ini adalah generator, yakni mesin berputar yang mengubah energi mekanis menjadi energi listrik. Perlu adanya sistem pemantauan yang dapat mengukur keluaran daya, suhu, arus, frekuensi dan tegangan yang dihasilkan oleh generator. Tujuan penelitian ini adalah untuk merancang dan membuat sebuah sistem pemantauan Generator dengan mengukur berbagai data yang dihasilkan oleh generator. Data hasil pengukuran tersebut dikirimkan ke sistem antarmuka pada personal computer melalui komunikasi serial. Sistem antarmuka menggunakan software Visual Basic 2010 Express, sedangkan pengolahan data menggunakan Mikrokontroler jenis Arduino Mega 2560. Hasil yang diperoleh antara lain sistem dapat menampilkan besaran tegangan, arus beban, daya aktif, daya reaktif, daya semu, factor daya, besaran suhu, dan frekuensi pada LCD dan juga dikirimkan ke sistem antarmuka pada pc melalui komunikasi serial. Penggunaan komunikasi Serial terkendala oleh banyaknya jumlah variabel data yang dikirim, semakin banyak jumlah variable data, kemungkinan error dalam pengiriman data semakin besar.

Kata kunci : Generator; Sistem Pemantauan; Visual Basic; Komunikasi Serial; Arduino

ABSTRACT

The power plant is part of industrial equipment used to generate electricity from various power sources, such as PLTU, PLTD, nuclear power plants, and hydropower plants. The main part of this power plant is a generator, which is a rotating machine that converts mechanical energy into electrical energy. The need for a monitoring system that can measure the output power, temperature, current, frequency and voltage generated by the generator. The purpose of this study was to design and create a generator monitoring system by measuring various data generated by the generator. The measurement data is sent to the interface system on personal computers through serial communication. System interface using Visual Basic 2010 Express software, while processing data using Microcontroller type Arduino Mega 2560. The results obtained include the system can display the amount of voltage, load current, active power, reactive power, apparent power, power factor, temperature, and frequency on the LCD and also sent to the system interface on the PC through serial communication. The use of Serial communication is constrained by the large number of data variables sent, the more the number of data variables, the chances of error in the data transmission the greater.

Keywords: Generator; Monitoring System; Visual Basic; Serial Communication; Arduino

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan laporan Proyek Akhir ini dengan baik. Adapun laporan Proyek Akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III (D-III) di Politeknik Manufaktur Negeri Bangka Belitung.

Dalam Proyek Akhir ini penulis membuat sebuah sistem pemantauan pada generator berbasis GUI (*Graphical User Interface*) dengan menggunakan *Arduino* dan *Visual Basic*. Penulis mengakui bahwa selesainya Proyek Akhir ini tidak lepas dari bantuan banyak pihak yang telah membantu dan memberi dukungan dalam membuat alat maupun dalam menyelesaikan laporan Proyek Akhir ini. Untuk itu penulis mengucapkan terima kasih kepada :

1. Orang tua serta keluarga yang selalu memberikan kasih sayang, doa serta dukungan.
2. Bapak Sugeng Ariyono, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Aan Febriansyah, M.T, selaku Kepala Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Eko Sulisty, M.T, selaku Kepala Prodi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Yudhi, M.T, selaku wali kelas III EA sekaligus pembimbing 1 dalam Proyek Akhir ini.
6. Bapak Indra Dwisaputra, M.T, selaku pembimbing 2 dalam Proyek Akhir ini.
7. Seluruh staf pengajar dan karyawan di Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan-rekan mahasiswa tingkat akhir Politeknik Manufaktur Negeri Bangka Belitung.

9. Teman-teman yang telah ikut mendukung dan memberikan bantuan serta masukan dalam pembuatan Proyek Akhir ini.
10. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa dalam laporan Proyek Akhir ini masih jauh dari sempurna, untuk itu penulis sangat mengharapkan semua jenis saran, kritik dan masukan yang bersifat membangun dalam rangka perbaikan laporan ini. Demikian laporan ini dibuat dan semoga laporan ini dapat bermanfaat dan menambah wawasan bagi pembaca. Akhir kata penulis mengucapkan terima kasih.

Sungailiat, 27 juni 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah dan Batasan Masalah	1
1.3 Tujuan Proyek Akhir.....	2
BAB II DASAR TEORI	3
2.1 <i>AC Digital Multifunction Meter PZEM-004T Module</i>	3
2.2 Sensor Frekuensi	4
2.3 Sensor Suhu LM35	5
2.4 <i>Arduino Mega 2560</i>	6
2.5 <i>Visual Basic</i>	7
2.6 <i>Liquid Crystal Display</i>	8
2.7 Teori Perhitungan Daya	9
2.8 Generator AC 1 Phasa.....	10
BAB III METODE PELAKSANAAN	11
3.1 Metode Pelaksanaan.....	11
3.2 Pengumpulan Data	12
3.3 Perancangan Perangkat Keras (hardware)	12

3.4	Perancangan Perangkat Lunak (software)	13
3.5	Pengembangan Alat dan Uji Coba.....	14
BAB IV PEMBAHASAN.....		15
4.1	Diagram Blok.....	15
4.2	<i>Hardware</i> Box Panel.....	16
4.2.1	Proses Pembuatan Box Panel.....	16
4.2.2	Perakitan Komponen Sensor.....	16
4.3	<i>Hardware</i> Elektrik	17
4.3.1	Rangkaian Catu Daya	17
4.3.1.1	Perancangan Rangkaian Catu Daya.....	17
4.3.1.2	Pembuatan Rangkaian Catu Daya.....	18
4.3.1.3	Pengujian Rangkaian Catu Daya	18
4.3.2	Rangkaian LCD 20x4	19
4.3.2.1	Perancangan Rangkaian LCD 20x4	19
4.3.2.2	Pembuatan Rangkaian LCD 20x4.....	20
4.3.2.3	Pengujian Rangkaian LCD 20x4	20
4.3.3	Arduino Mega 2560	22
4.3.3.1	Perancangan Arduino Mega 2560.....	22
4.3.3.2	Perakitan Arduino Mega 2560.....	22
4.3.4	Modul Sensor PZEM-004T.....	23
4.3.4.1	Perancangan Modul Sensor PZEM-004T	23
4.3.4.2	Perakitan Modul Sensor PZEM-004T	23
4.3.4.3	Pengujian Modul Sensor PZEM-004T.....	24
4.3.5	Sensor Frekuensi.....	27
4.3.5.1	Perancangan Sensor Frekuensi	27
4.3.5.2	Pembuatan Sensor Frekuensi	28
4.3.5.3	Pengujian Sensor Frekuensi.....	29
4.3.6	Sensor Suhu	31
4.3.6.1	Perancangan Sensor Suhu.....	31
4.3.6.2	Pembuatan Sensor Suhu.....	32
4.3.6.3	Pengujian Sensor Suhu	32

4.4	Sistem <i>Interface</i>	34
4.4.1	Visual Basic	34
4.4.1.1	Perancangan <i>Software</i> Sistem <i>Interface</i>	34
4.4.1.2	Pembuatan <i>Software</i> Sistem <i>Interface</i>	35
4.4.1.3	Pengujian <i>Software</i> Sistem <i>Interface</i>	37
4.5	Pengujian Keseluruhan Alat	38
BAB V KESIMPULAN DAN SARAN		43
5.1	Kesimpulan	43
5.2	Saran	43
DAFTAR PUSTAKA		44
LAMPIRAN		46

DAFTAR TABEL

Tabel	Halaman
2.1 Spesifikasi <i>PZEM-004T module</i>	3
2.2 Konfigurasi Pin Modul <i>PZEM-004T</i>	4
4.1 Hasil Pengujian Rangkaian Catu Daya	19
4.2 Pengujian Tegangan.....	25
4.3 Pengujian Arus.....	26
4.4 Pengujian Daya.....	27
4.5 Pengujian Sensor Frekuensi.....	30
4.6 Pengujian Sensor Suhu	34
4.7 Hasil Pengujian Arus	40
4.8 Hasil Pengujian Tegangan	41
4.9 Hasil Pengujian Daya	41

DAFTAR GAMBAR

Gambar	Halaman
2.1 <i>AC digital multifunction meter PZEM-004T module</i>	3
2.2 Bentuk Fisik Modul Sensor Tegangan ZMPT101B	4
2.3 Bentuk Fisik Modul Sensor Arus ACS712.....	5
2.4 Skematik rangkaian op-amp.	5
2.5 Bentuk Fisik Modul Sensor Suhu LM35	6
2.6 Bentuk Fisik Arduino Mega 2560	6
2.7 <i>LCD (Liquid Crystal Display) 4x20</i>	8
2.8 Skematik dari modul I2C LCD.....	9
2.9 Bentuk Fisik modul I2C LCD	9
2.10 Bentuk Fisik Generator.....	10
3.1 <i>Flow Chart</i> Proses Pengerjaan Proyek Akhir.....	11
3.2 Blok Diagram <i>Hardware</i> Secara Umum	12
3.3 <i>Flow Chart</i> Perancangan Perangkat Lunak	14
4.1 Diagram Blok.....	15
4.2 Bentuk Fisik Junction Box	16
4.3 Konstruksi Box Panel	17
4.4 Adaptor 5VDC 3A.....	17
4.5 Skematik Rangkaian Catu Daya	18
4.6 Rangkaian Catu Daya	18
4.7 Pengukuran Tegangan Catu Daya 5V DC	19
4.8 Skematik Rangkaian LCD 20x4	20
4.9 Rangkaian LCD 20x4	20
4.10 Hasil Pengujian LCD 20x4.....	21
4.11 Skematik Arduino Mega 2560.....	22
4.12 Pemasangan Arduino Mega 2560.....	22
4.13 Skematik Rangkaian PZEM-004T.....	23

4.14	Penempatan Modul PZEM-004T Pada Box Panel	23
4.15	Pengujian Tegangan.....	25
4.16	Pengujian Arus.....	26
4.17	Pengujian Daya.....	27
4.18	Skematik Rangkaian Sensor Frekuensi	28
4.19	Sinyal Keluaran Op-amp	28
4.20	Penempatan Sensor Frekuensi	29
4.21	Pengujian Sensor Frekuensi.....	30
4.22	Skematik Sensor LM35DZ.....	31
4.23	Pemasangan Sensor Suhu	32
4.24	Pengujian Sensor Suhu.	33
4.25	Tampilan Pengujian Sensor Suhu.....	33
4.26	Membuat <i>project</i> baru pada <i>visual basic</i>	35
4.27	Desain <i>layout</i> form1.	36
4.28	Hasil Pengujian Pembuatan Sistem <i>Interface</i>	37
4.29	Hasil Pengujian Arus dan Tegangan Input	38
4.30	Hasil Pengujian Sistem Interface.....	39
4.31	Hasil Pengujian Daya dan Tegangan..	39
4.32	Hasil Pengujian Arus dan Tegangan.....	39
4.33	Hasil Pengujian Daya Pada LCD.....	40
4.34	Hasil Pengujian Arus dan Tegangan Pada LCD.....	40

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

Lampiran 2 : Program Arduino Mega 2560: Sistem Pemantauan Generator Berbasis *Arduino* dan *Visual Basic*

Lampiran 5 : Program Visual Basic: Sistem Pemantauan Generator Berbasis *Arduino* dan *Visual Basic*

Lampiran 4 : Desain *Layout* Visual Basic: Sistem Pemantauan Generator Berbasis *Arduino* dan *Visual Basic*

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pembangkit listrik adalah bagian dari alat industri yang dipakai untuk membangkitkan tenaga listrik dari berbagai sumber tenaga, seperti PLTU, PLTD, PLTN, PLTA, dan lain-lain. Bagian utama dari pembangkit listrik ini adalah generator, yakni mesin berputar yang mengubah energi mekanis menjadi energi listrik dengan menggunakan prinsip medan magnet dan penghantar listrik [1].

Pada umumnya, generator konvensional hanya memiliki alat ukur tegangan saja, padahal sinyal keluaran generator tersebut memiliki beberapa variabel data seperti arus beban, daya aktif, daya reaktif, serta frekuensi keluaran generator. Karena terbatasnya alat ukur pada generator maka operator harus membawa banyak alat ukur agar bisa melakukan pengukuran pada generator. Hal ini menyebabkan kurangnya efisiensi waktu dalam pengukuran. Karena itu, perlu adanya sistem pemantauan yang dapat mengukur keluaran daya, suhu, arus, frekuensi dan tegangan yang dihasilkan oleh generator sekaligus. Untuk mempermudah operator pada saat melakukan pengukuran dan pengambilan data pada generator.

Pada penelitian ini penulis mengangkat tema berjudul sistem pemantauan generator berbasis arduino dan *visual basic*. Dengan menerapkan sistem GUI (*Graphical User Interface*) agar memudahkan operator untuk memantau keadaan generator serta hasil keluaran generator tersebut.

1.2 Perumusan Masalah dan Batasan Masalah

Rumusan masalah pada tugas akhir ini adalah :

- a. Bagaimana cara merancang dan membuat sistem pemantauan generator berbasis arduino dan *visual basic*.

- b. Bagaimana menguji sistem pemantauan generator berbasis arduino dan *visual basic*.

Sedangkan batasan masalahnya adalah :

- a. Sistem ini hanya bisa diakses menggunakan kabel *serial*.
- b. Pengujiannya dilakukan menggunakan generator 1 *Phase* dengan tegangan keluaran 220V.
- c. Dapat mengukur tegangan beban 250 VAC dengan arus maksimal 5A.
- d. Data pemantauan ditampilkan ke grafik LCD dan ke satu *Personal Computer*.

1.3 Tujuan Proyek Akhir

Dengan mengacu latar belakang di atas maka tugas akhir ini bertujuan:

1. Cara merancang dan membuat sistem pemantauan generator berbasis arduino dan *visual basic*.
2. Menguji sistem pemantauan generator berbasis arduino dan *visual basic*.

BAB II DASAR TEORI

2.1 *AC Digital Multifunction Meter PZEM-004T Module*

Modul PZEM-004T adalah sebuah modul sensor multifungsi yang berfungsi untuk mengukur daya, tegangan, arus dan energi yang terdapat pada sebuah aliran listrik. Modul ini sudah dilengkapi sensor tegangan dan sensor arus (*CT*) yang sudah terintegrasi. Dalam penggunaannya, alat ini khusus untuk penggunaan dalam ruangan (*indoor*) dan beban yang terpasang tidak diperbolehkan melebihi daya yang sudah ditetapkan. Dimensi fisik dari modul PZEM-004T ini adalah sebuah papan yang berukuran panjang 7.4 cm dengan lebar 3,1 cm. Data sensor yang didapat selanjutnya dikirim ke *arduino board* melalui komunikasi serial di pin RX TX. Berikut gambar dari sensor *PZEM-004T*.



Gambar 2.1 Bentuk fisik *AC digital multifunction meter PZEM-004T module* [2]

Tabel 2.1 Spesifikasi *PZEM-004T module*

No.	Spesifikasi	Keterangan
1.	Tegangan Kerja	80 ~ 260VAC
2.	Arus Pengukuran	0 ~ 100A
3.	Daya	22kW
4.	Frekuensi	45 – 65Hz

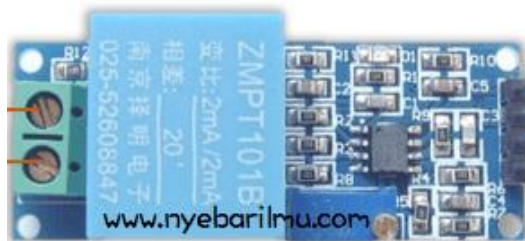
Konfigurasi pin modul *PZEM-004T*

Tabel 2.2 Konfigurasi Pin Modul *PZEM-004T*

No.	Pin	Fungsi
1.	<i>VCC</i>	Pin ini berhubungan dngan tegangan 5 volt
2.	<i>RX</i>	Komunikasi Serial TTL ke Arduino(TX)
3.	<i>TX</i>	Komunikasi Serial TTL ke Arduino(RX)
4.	<i>GND</i>	Pin ini berhubungan dngan tegangan 0 volt (<i>Ground</i>)

2.2 Sensor Frekuensi

Sensor ini menggunakan modul ACS712 dan modul ZMPT101B untuk mengkondisikan sinyal AC. Kedua modul sensor ini digunakan sebagai pengganti dari transformator untuk menurunkan tegangan AC sehingga sinyalnya dapat dibaca oleh arduino, karena pin arduino hanya mengeluarkan atau membaca tegangan 5V. Bentuk fisik sensor tegangan ZMPT101B dapat dilihat pada gambar dibawah ini.



Gambar 2.2 Bentuk fisik modul sensor tegangan ZMPT101B[3]

Modul sensor ini memerlukan catu daya sebesar 5Vdc. Hasil sinyal keluarannya berupa sinyal analog. Sensor ini dapat mengukur tegangan AC satu fasa dari 110V hingga 250V, sedangkan input arus sensor ini hanya sebesar 2mA.

ACS712 adalah sensor arus yang bekerja berdasarkan efek medan. Sensor arus ini dapat digunakan untuk mengukur arus AC atau DC. Modul sensor ini telah dilengkapi dengan rangkaian penguat *operasional*, sehingga *sensitivitas* pengukuran arusnya meningkat dan dapat mengukur perubahan arus yang kecil.

Spesifikasi Sensor Arus ACS712:

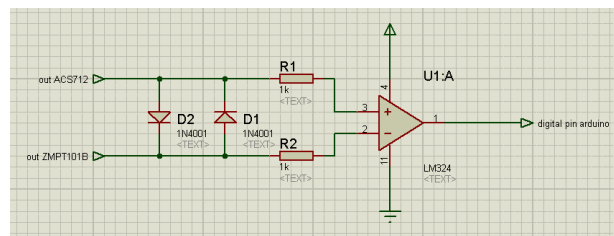
- *Supply Voltage:* 4.5V~5.5V DC
- *Measure Current Range:* -5A~ 5A
- *Sensitivity:* 180mV/A ~190mV/A, *Typical:* 185mV/A

Berikut adalah bentuk fisik dari modul sensor arus ACS712 yang ditunjukkan pada gambar 2.3.



Gambar 2.3 Bentuk Fisik modul sensor arus ACS712 [4]

Kemudian sinyal *output* dari kedua modul tersebut dibandingkan oleh sebuah op-amp untuk menghasilkan sinyal pulsa agar pembacaan periode yang dibutuhkan dalam satu gelombang lebih mudah. Berikut adalah skematik dari rangkaian op-amp tersebut yang ditunjukkan pada gambar 2.4.



Gambar 2.4 Skematik rangkaian op-amp.

2.3 Sensor Suhu LM35

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika yang diproduksi oleh *National Semiconductor*. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.

Arduino tidak perlu perangkat *chip programmer* karena didalamnya sudah ada *bootloader* yang akan menangani *upload* program dari komputer. Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap. Bahasa pemrograman yang digunakan adalah bahasa C.

Arduino Mega 2560 memiliki berbagai pin yang cocok digunakan dalam proyek akhir ini. Pin Vin menerima tegangan *input* ke papan menggunakan sumber daya eksternal. Pin 5V berfungsi untuk memberikan *output* 5V yang diatur dari *regulator on-board*. Pin GND berhubungan dengan tegangan 0 volt (*Ground*). Beberapa pin *serial* yang digunakan dalam pengiriman data dari sensor yaitu *Serial 0*: 0(RX) dan 1(TX); *Serial 1*: 19(RX) dan 18 (TX); *Serial 2*: 17(RX) dan 16(TX) dan *Serial 3*: 15(RX) dan 14(TX).

2.5 *Visual Basic*

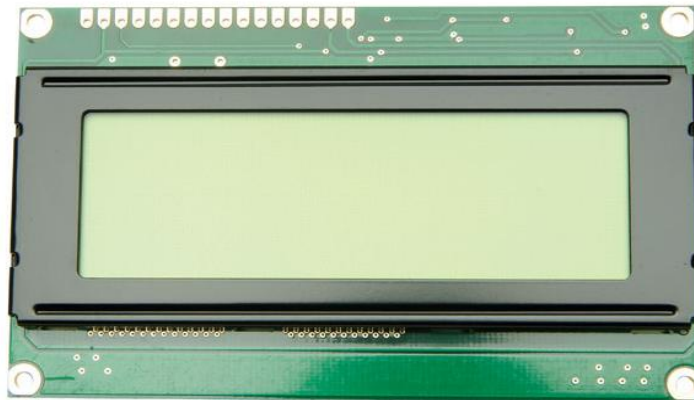
Visual Basic merupakan bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Basic* selain disebut sebagai bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program aplikasi berbasis windows. *Visual Basic* memungkinkan pembuatan aplikasi *Graphical User Interface (GUI)* atau pemrograman yang menggunakan tampilan grafis sebagai alat komunikasi dengan pemakainya. Dalam *Visual Basic* untuk pembuatan tampilan user interface relatif mudah dilakukan karena hanya perlu meletakkan objek-objek grafis ke sumber (*form*) yang sudah disediakan *Visual Basic*. Setelah itu cukup mengatur properti dari objek-objek tersebut.

Visual Basic mempunyai beberapa kemampuan atau manfaat yaitu sebagai berikut:

1. Untuk membuat program aplikasi berbasis *Windows*.
2. Untuk membuat *ActiveX*, Aplikasi internet dan lain sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran EXE yang bersifat *executable* atau dapat langsung dijalankan.[8]

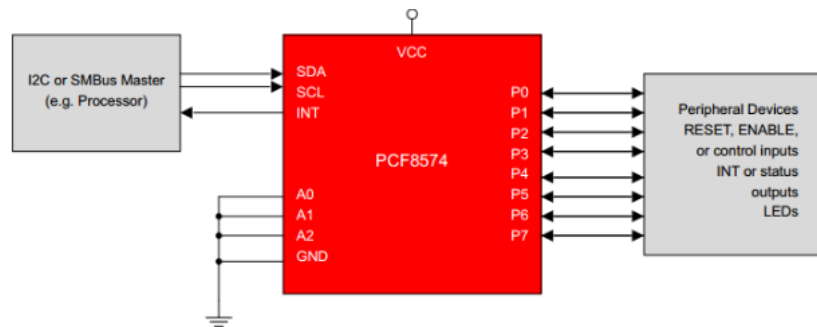
2.6 *Liquid Crystal Display*

LCD (*Liquid Crystal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi *CMOS logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD berfungsi sebagai penampil data baik dalam bentuk karakter, huruf dan angka. Dalam modul LCD terdapat Mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan register. LCD yang digunakan dapat dilihat pada gambar 2.7 berikut.



Gambar 2.7 LCD (*Liquid Crystal Display*) 4x20 [10]

Untuk menghemat penggunaan pin pada arduino, maka digunakanlah modul I2C LCD. LCD modul ini adalah *display hardware* yang paling umum digunakan dalam *project* arduino dan jenis mikrokontroler lainnya karena dapat menghemat pin arduino *board*. Modul ini mempunyai 16 pin *output* yang dapat dihubungkan dengan pin LCD 20x4 dan 16x2 secara langsung dan memiliki 4 pin input yaitu VCC, GND, SDA, dan SCL. Berikut adalah skematik dari modul I2C LCD yang ditunjukkan pada gambar 2.8.



Gambar 2.8 Skematik dari modul I2C LCD[11]

Adapun bentuk fisik dari modul I2C LCD dapat dilihat dari gambar 2.9 berikut.



Gambar 2.9 Bentuk Fisik modul I2C LCD [12]

2.7 Teori Perhitungan Daya

Daya adalah energi yang dikeluarkan untuk melakukan usaha. Dalam sistem tenaga listrik, daya merupakan jumlah energi yang digunakan untuk melakukan kerja atau usaha. Daya aktif (*Active Power*) adalah daya yang terpakai untuk melakukan energi sebenarnya. Daya listrik biasanya dinyatakan dalam satuan *Watt* atau *Horsepower (HP)*, Daya dinyatakan dalam P, Tegangan dinyatakan dalam V dan Arus dinyatakan dalam I, sehingga besarnya daya dinyatakan :

$$P = V \cdot I \cdot \cos \varphi \dots \dots \dots (2.1)$$

Daya ini digunakan secara umum oleh konsumen dan dikonversikan dalam bentuk kerja.

Daya reaktif adalah jumlah daya yang diperlukan untuk pembentukan medan magnet. Dari pembentukan medan magnet maka akan terbentuk fluks

medan magnet. Contoh daya yang menimbulkan daya reaktif adalah transformator, motor, lampu pijar dan lain – lain. Satuan daya reaktif adalah Var.

$$Q = V.I.\sin \varphi \dots\dots\dots(2.2)$$

Daya nyata (*Apparent Power*) adalah daya yang dihasilkan oleh perkalian antara tegangan rms dan arus rms dalam suatu jaringan atau daya yang merupakan hasil penjumlahan trigonometri daya aktif dan daya reaktif. Satuan daya nyata adalah VA.

$$S = V.I.\dots\dots\dots(2.3)$$

Faktor daya (*Cos φ*) dapat didefinisikan sebagai rasio perbandingan antara daya aktif (*Watt*) dan daya nyata (VA) yang digunakan dalam sirkuit AC atau beda sudut fasa antara V dan I yang biasanya dinyatakan dalam *cos φ* [15].

$$\begin{aligned} \text{Faktor Daya} &= \text{Daya Aktif (P) / Daya Nyata (S)} \\ &= \text{kW / kVA} \\ &= V.I \cos \varphi / V.I \\ &= \cos \varphi \dots\dots\dots(2.4) \end{aligned}$$

2.8 Generator AC 1 Phasa

Generator merupakan mesin yang merubah energi mekanik menjadi energi listrik. Generator yang digunakan pada proyek akhir kali ini adalah MOTOYAMA SPG8800E2, yaitu sebagai objek yang akan diukur untuk pengujian alat pemantauan generator berbasis Arduino dan *visual basic*. Generator ini mengeluarkan listrik 1 phasa dengan daya 8800 watt, tegangan 220-250 volt dan frekuensi 50 Hz. Adapun bentuk fisik dari generator yang kami gunakan ditunjukkan pada gambar 2.10.

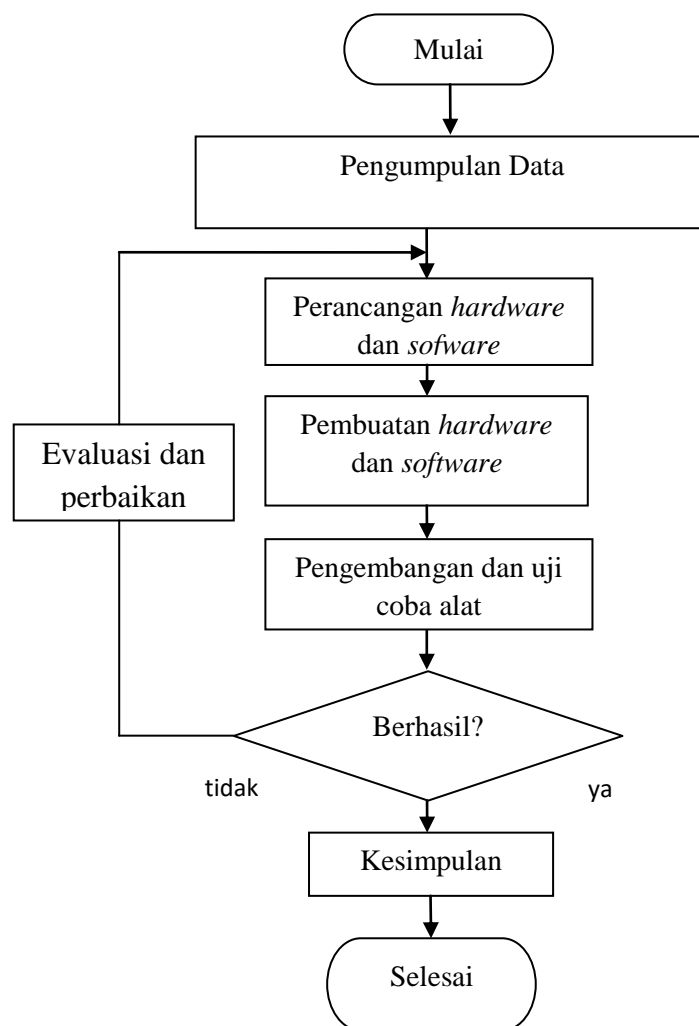


Gambar 2.10 Bentuk Fisik Generator

BAB III METODE PELAKSANAAN

3.1 Metode Pelaksanaan

Pada bab ini metode pelaksanaan dibagi menjadi dua bagian: pembuatan perangkat keras (*hardware*) dan perancangan (*software*). Berikut *flow chart* metode pelaksanaan penelitian yang ditunjukkan pada gambar dibawah ini.



Gambar 3.1 *flow chart* proses pengerjaan proyek akhir

3.2 Pengumpulan Data

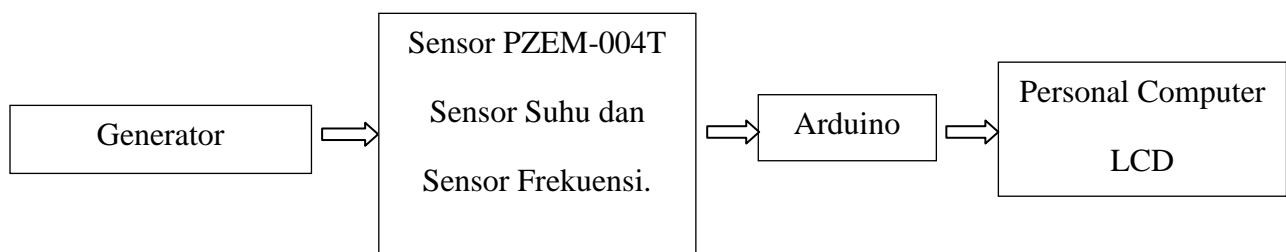
Survei dan pengumpulan data dilakukan untuk mencari data-data sumber referensi yang diperlukan guna mengetahui dan mengatasi permasalahan yang terjadi. Pada tahap ini dilakukan pengamatan mengenai alat yang akan dirancang melalui referensi dari beberapa sumber seperti literatur buku, jurnal, konsultasi, *browsing* dan lain sebagainya. Setelah data-data dikumpulkan selanjutnya data tersebut diolah dan dirangkum. Dari proses pengolahan data didapat daftar kebutuhan dari alat yang akan dibuat pada Proyek Akhir. Data-data tersebut kemudian akan menjadi acuan dalam merancang sistem *monitoring*.

3.3 Perancangan Perangkat Keras (*hardware*)

Perancangan *hardware* meliputi perancangan alat dan kontrol yang akan diterapkan. Pada tahap ini dibuat rancangan berupa *hardware* yang akan digunakan dalam pembuatan Proyek Akhir. Rancangan ini bertujuan untuk membuat proses pembuatan Proyek Akhir sistem *monitoring* menjadi lebih cepat dan sesuai dengan target yang diinginkan.

Perancangan *hardware* yang dilakukan meliputi perancangan elektrik dari sistem *monitoring*. Perancangan elektrik sistem *monitoring* yaitu perancangan rangkaian kontrol serta komponen elektrik yang akan dibutuhkan dalam pembuatan Proyek Akhir sistem *monitoring*.

Generator sebagai objek atau alat yang akan diukur keluarannya berupa besaran listrik yang akan diberi beban rumahan. Dan LCD sebagai alat yang akan menampilkan hasil pengukuran pada generator. Berikut adalah Blok diagram *hardware* secara umum yang ditunjukkan pada gambar 3.2.



Gambar 3.2 Blok diagram *hardware* secara umum

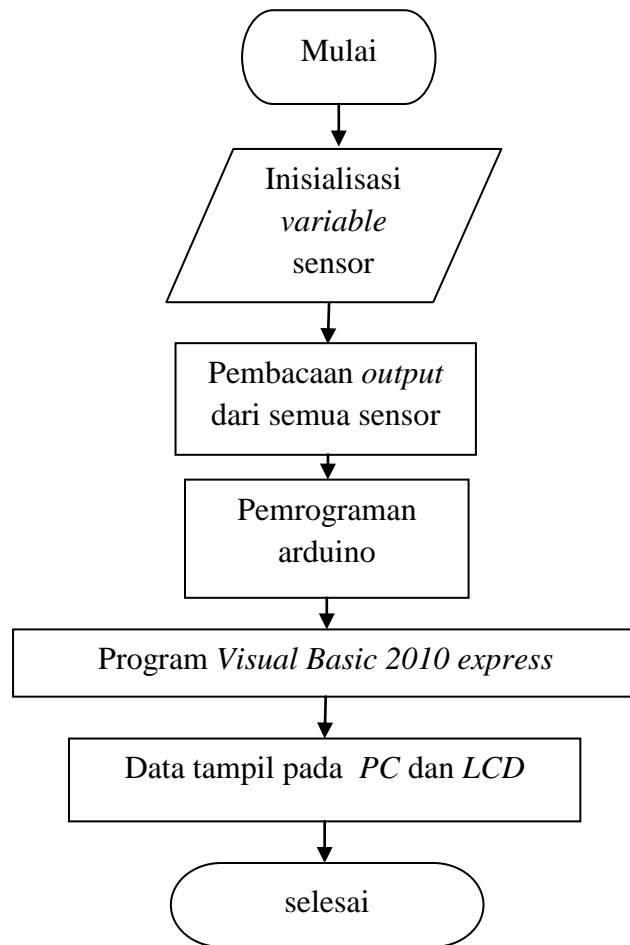
3.4 Perancangan Perangkat Lunak (*software*)

Adapun proses metodologi rancangan perangkat lunak dimulai dari inialisasi *variable* sensor, pembacaan *output* dari semua sensor, pemrograman pada arduino, pemrograman pada *visual basic* dan tampilan pada LCD dan *visual basic*.

Inialisasi *variable* sensor ialah pengenalan terhadap sensor yang digunakan untuk mengukur *output* generator. Pembacaan *output* dari semua sensor adalah bagaimana membaca data *output* dari semua sensor dengan menggunakan arduino.

Pemrograman pada arduino merupakan tahap yang penting karena arduino berfungsi sebagai kontrol sehingga data sensor masuk ke arduino untuk diolah. Pemrograman *visual basic* adalah sebuah program yang dibuat sebagai *interface*, tahap ini juga sama pentingnya dengan pemrograman arduino karena saling berhubungan satu sama lain.

Tampilan di LCD dan *visual basic* ini merupakan tahap terakhir dengan menggabungkan satu sama lain antara program *visual basic* dan program arduino sehingga menjadi satu kesatuan. Berikut adalah *Flow chart* perancangan perangkat lunak yang ditunjukkan pada gambar 3.3.



Gambar 3.3 *Flow chart* perancangan perangkat lunak

3.5 Pengembangan Alat dan Uji Coba

Jika pembuatan *hardware* dan *software* sudah selesai, maka dilakukan pengembangan dan uji coba alat yang dibuat. Uji coba ini berfungsi untuk memastikan apakah kontrol yang digunakan sudah benar dan mendukung *hardware* yang telah dibuat agar dapat mencapai target yang diinginkan.

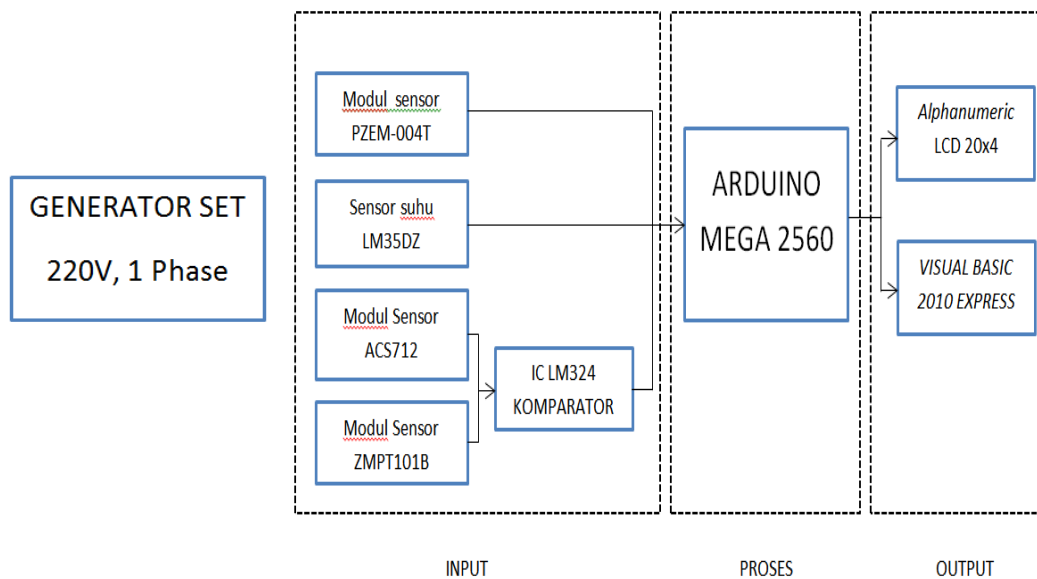
Apabila sistem kerja alat tidak sesuai yang diinginkan maka dilakukan evaluasi dan perbaikan sistem kerja alat. Jika berhasil, maka dilanjutkan kembali ke proses berikutnya.

BAB IV PEMBAHASAN

Pada bab ini akan dibahas mengenai proses atau perancangan desain dan pengembangan data uji coba peralatan-peralatan yang dikembangkan dalam pembuatan Proyek Akhir yang berjudul “Sistem Pemantauan Generator Berbasis *Arduino* dan *Visual Basic*”, yaitu sebagai berikut.

4.1 Diagram Blok

Diagram blok dari Sistem Pemantauan Generator Berbasis *Arduino* dan *Visual Basic* ditunjukkan pada gambar 4.1 berikut.



Gambar 4.1 Diagram Blok

Pada blok *input*, modul sensor PZEM-004T membaca data tegangan, arus dan daya. Modul sensor ZMPT101B dan Modul ACS712 membaca sinyal frekuensi AC yang kemudian dibandingkan dengan IC komparator LM324. Sensor LM35DZ mengubah temperatur pada body generator menjadi data analog.

Pada blok proses, seluruh data tersebut dikirimkan ke arduino, kemudian data yang dikirimkan diolah oleh arduino sehingga pada blok *output*, LCD 20x4 menampilkan nilai tegangan, arus, daya aktif, daya reaktif, daya semu, faktor daya, frekuensi Sinyal AC dan temperatur pada generator. Data yang ditampilkan pada LCD juga dikirimkan ke sistem *interface* menggunakan *software visual basic* melalui komunikasi serial, sehingga sistem *interface* juga menampilkan seluruh data yang ada pada LCD.

4.2 Hardware Box Panel

4.2.1 Proses Pembuatan Box Panel

Proses pembuatan *box* panel diputuskan dengan membeli *junction box* yang telah jadi dan biasa dijual secara umum. Bentuk fisik *junction box* dapat dilihat pada gambar 4.2 berikut.

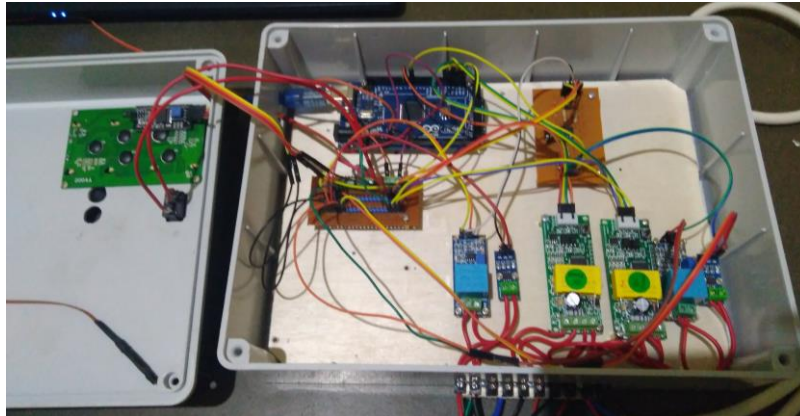


Gambar 4.2 Bentuk Fisik *Junction Box*

Pembuatan *box* kemudian dilanjutkan dengan pemotongan bahan triplek untuk *base plate* dudukan komponen sesuai dengan hasil rancangan yang dibuat sebelumnya.

4.2.2 Perakitan Komponen Sensor

Setelah bagian *box* panel dari *base plate* selesai dibangun selanjutnya adalah penempatan komponen menjadi kesatuan bentuk yang utuh. Pemasangan konstruksi dilakukan menggunakan komponen *spacer*, baut dan mur untuk menempatkan seluruh komponen sensor ke *base plate*. Berikut adalah gambar konstruksi *box* panel yang telah jadi ditunjukkan pada gambar 4.3.



Gambar 4.3 Konstruksi Box Panel

4.3 *Hardware* Elektrik

4.3.1 Rangkaian Catu Daya

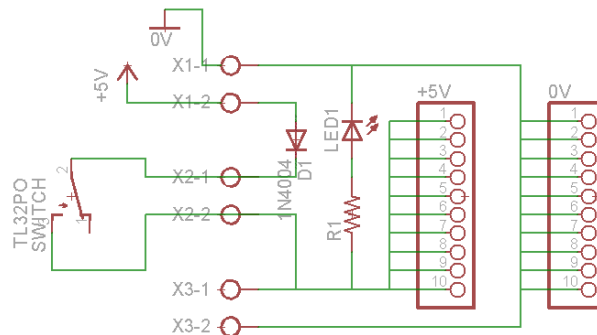
4.3.1.1 Perancangan Rangkaian Catu Daya

Pada proyek akhir ini rangkaian catu daya yang digunakan adalah adaptor 5VDC 3A. Tegangan *output* yang diinginkan yaitu 5V DC untuk Mikrokontroler Arduino Mega 2560 dan komponen lainnya seperti LCD 20x4, modul sensor PZEM-004T, modul sensor ACS712, modul sensor ZMPT101B, dan sensor suhu LM35DZ. Berikut adalah adaptor yang dapat dilihat pada gambar 4.4.



Gambar 4.4 Adaptor 5VDC 3A

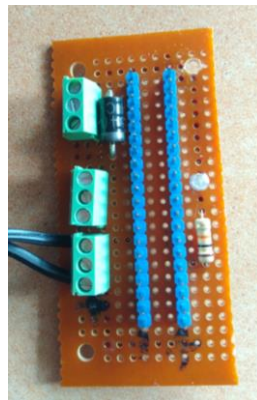
Adapun skematik dari rangkaian catu daya yang digunakan ditunjukkan pada gambar 4.5 berikut.



Gambar 4.5 Skematik Rangkaian Catu Daya

4.3.1.2 Pembuatan Rangkaian Catu Daya

Pembuatan rangkaian catu daya dilakukan berdasarkan hasil rancangan skematik yang telah dibuat. Proses pembuatan rangkaian catu daya ini dimulai dari pembuatan pemasangan komponen hingga rangkaian penyolderan. Berikut adalah rangkaian catu daya yang telah jadi ditunjukkan pada gambar 4.6.



Gambar 4.6 Rangkaian Catu Daya

4.3.1.3 Pengujian Rangkaian Catu Daya

Uji coba rangkaian catu daya dilakukan untuk menguji apakah rangkaian catu daya yang dibuat sesuai dengan tegangan kerja dari sensor dan mikrokontroler yang digunakan. Pada proyek akhir ini kita menggunakan tegangan catu daya 5Vdc yang cocok dengan Arduino dan sensor yang digunakan. Berikut adalah gambar hasil pengujian rangkaian catu daya yang dibuat ditunjukkan pada gambar 4.7.



Gambar 4.7 Pengukuran Tegangan Catu Daya 5V DC

Hasil pengujian rangkaian catu daya ditunjukkan pada tabel 4.1 berikut.

Tabel 4.1 Hasil Pengujian Rangkaian Catu Daya

Teori	Hasil Pengujian	Presentase Error
5 V DC	4,9 V DC	2 %

Hasil perhitungan persentase *error* adalah sebagai berikut.

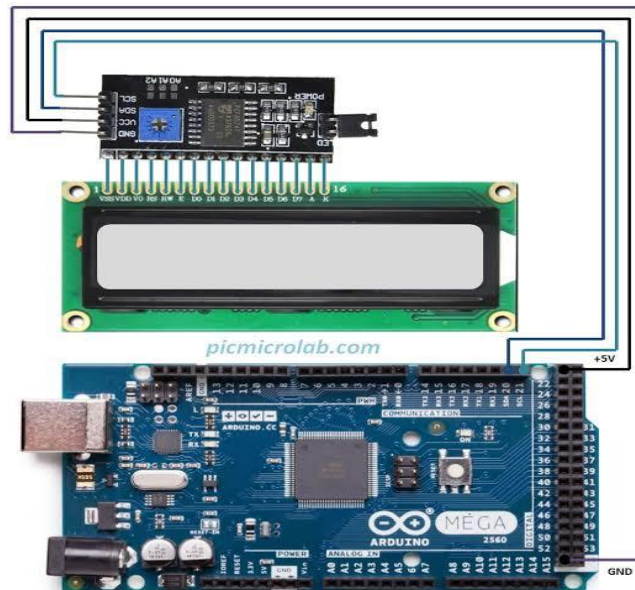
$$\text{Persentase error} = \left| \frac{5-4,9}{5} \right| \times 100\% = 2\%$$

Persentase *error* sebesar 2% ini masih termasuk dalam toleransi alat yaitu $\pm 5\%$.

4.3.2 Rangkaian LCD 20x4

4.3.2.1 Perancangan Rangkaian LCD 20x4

Rangkaian LCD pada proyek akhir ini menggunakan komunikasi I²C LCD modul. Dengan menggunakan pin SDA dan SCL pada arduino untuk menghemat penggunaan pin arduino. LCD 20x4 berfungsi sebagai tampilan dari variable-variabel yang diukur. Gambar skematik dari rangkaian LCD 20x4 ditunjukkan pada gambar 4.8 berikut.



Gambar 4.8 Skematik Rangkaian LCD 20x4[13]

4.3.2.2 Pembuatan Rangkaian LCD 20x4

Proses pembuatan rangkaian LCD 20x4 ini dilakukan dengan menyolder modul LCD I2C ke LCD 20x4 sesuai dengan skematik yang telah dibuat. Adapun rangkaian LCD 2x16 yang dibuat ditunjukkan pada gambar 4.9 berikut.



Gambar 4.9 Rangkaian LCD 20x4

4.3.2.3 Pengujian Rangkaian LCD 20x4

Pengujian rangkaian LCD 20x4 dimulai dari pemasangan pin modul I²C LCD pada Arduino Mega 2560 seperti yang ditunjukkan pada gambar 4.14 yaitu sebagai berikut:

- Pin GND diberikan *Ground*
- Pin VCC diberikan tegangan 5V

- Pin SDA terhubung ke SDA arduino
- Pin SCL terhubung ke SCL arduino

Pengujian dilakukan menggunakan pemrograman Arduino dengan *list* program sebagai berikut.

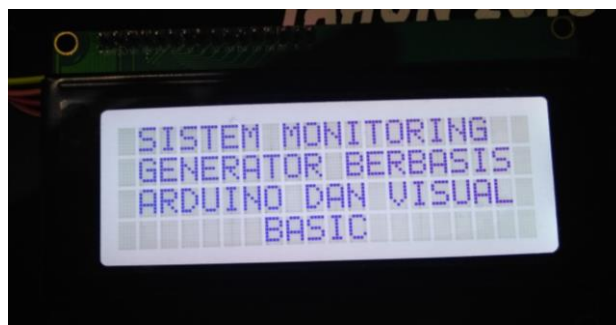
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,POSITIVE);
void setup() {
  lcd.begin(20,4);
}
void loop() {
  lcd.setCursor(0,0);
  lcd.print(" SISTEM MONITORING ");

  lcd.setCursor(0,1);
  lcd.print(" GENERATOR BERBASIS");

  lcd.setCursor(0,2);
  lcd.print(" ARDUINO DAN VISUAL");

  lcd.setCursor(0,3);
  lcd.print("          BASIC");}
```

Berikut adalah gambar hasil pengujian rangkaian LCD 20x4 menggunakan pemrograman Arduino dapat dilihat pada gambar 4.10.



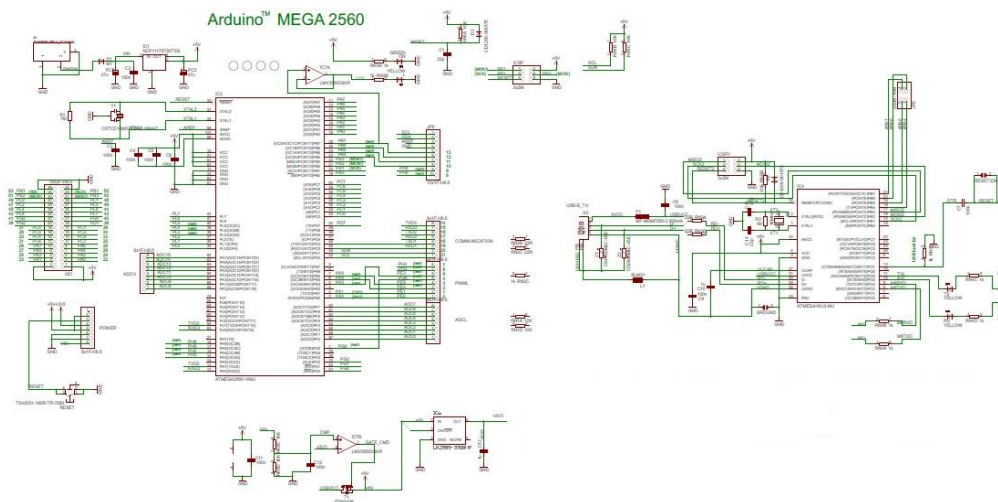
Gambar 4.10 Hasil Pengujian LCD 20x4

Dari hasil pengujian LCD20x4 dapat dianalisa bahwa rangkaian LCD20x4 yang dibuat dapat digunakan untuk menampilkan nilai berdasarkan program yang telah dibuat sehingga disimpulkan bahwa rangkaian LCD20x4 yang dibuat sesuai dengan rancangan yang diinginkan.

4.3.3 Arduino Mega 2560

4.3.3.1 Perancangan Arduino Mega 2560

Arduino Mega 2560 digunakan sebagai Mikrokontroler untuk mengontrol seluruh sistem. Arduino Mega 2560 dipilih karena memiliki fitur yang lebih tinggi dibandingkan jenis Arduino yang lainnya. Adapun skematik dari Arduino Mega 2560 dapat dilihat pada gambar 4.11 berikut.



Gambar 4.11 Skematik Arduino Mega 2560[14]

4.3.3.2 Perakitan Arduino Mega 2560

Perakitan Arduino Mega 2560 diputuskan dengan membeli modul Arduino Mega 2560 yang telah jadi dan biasa dijual secara umum. Pemasangan Arduino Mega 2560 pada box panel dapat dilihat pada gambar 4.12 berikut.

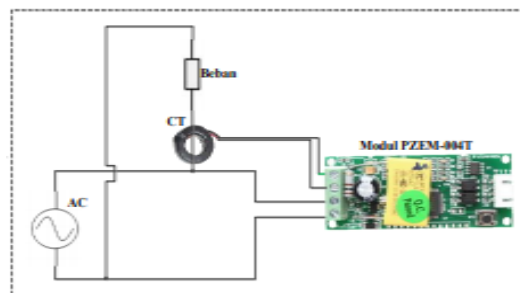


Gambar 4.12 Pemasangan Arduino Mega 2560

4.3.4 Modul Sensor PZEM-004T

4.3.4.1 Perancangan Modul Sensor PZEM-004T

Modul sensor PZEM-004T dipilih karena memiliki fitur yang lebih banyak dibandingkan dengan modul sensor tegangan dan arus lainnya. Tidak hanya mengukur besaran tegangan dan arus, sensor ini juga dapat mengukur nilai daya aktif, sehingga dengan menggunakan perhitungan maka didapatkan nilai daya semu, daya reaktif dan faktor daya. Modul ini juga cocok digunakan dengan mikrokontroler Arduino Mega 2560 karena sama-sama memiliki pin RX,TX yang merupakan pin komunikasi serial. Berikut adalah rangkaian skematik dari modul sensor PZEM-004T yang ditunjukkan pada gambar 4.13 dibawah ini.



Gambar 4.13 Skematik Rangkaian PZEM-004T[15]

4.3.4.2 Perakitan Modul Sensor PZEM-004T

Perakitan modul sensor PZEM-004T diputuskan dengan membeli modul sensor PZEM-004T yang telah jadi dan dijual secara umum. Pemasangan modul sensor pada box panel dapat dilihat pada gambar 4.14 berikut.



Gambar 4.14 Penempatan Modul PZEM-004T Pada Box Panel.

4.3.4.3 Pengujian Modul Sensor PZEM-004T

Pengujian rangkaian Modul Sensor PZEM-004T dimulai dari pemasangan pin output modul sensor pada Arduino Mega 2560 sebagai berikut:

- Pin GND diberikan *Ground*
- Pin VCC diberikan tegangan 5V
- Pin RX terhubung ke TX arduino
- Pin TX terhubung ke RX Arduino

Pengujian dilakukan menggunakan pemrograman Arduino dengan *list* program sebagai berikut.

```
#include <SoftwareSerial.h>
#include <PZEM004T.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

PZEM004T pzem(&Serial1);
IPAddress ip(192,168,1,1);

void setup()
{
  Serial.begin(9600);
  pzem.setAddress(ip);
  lcd.begin(20,4);
  lcd.backlight();
}

void loop()
{
  float v = pzem.voltage(ip);
  if (v < 0.0) v = 0.0;
  Serial.print(v);Serial.print("V; ");
  lcd.setCursor(0,1);
  lcd.print("V : ");
  lcd.print(v);
  lcd.setCursor(12,1);
  lcd.print("Volt ");
  Serial.println();
}
```

Adapun hasil pengujian dari sensor PZEM-004T dengan beban *rice cooker* ditunjukan pada gambar dibawah ini.

- Pengujian Tegangan

Hasil pengujian tegangan ditunjukan pada gambar 4.15 berikut.



Gambar 4.15 Pengujian Tegangan.

Hasil pengujian tegangan ditunjukkan pada tabel 4.2 berikut.

Tabel 4.2 Pengujian Tegangan

Pengujian	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase <i>Error</i> %
1	224,1 V	227,6 V	3.9
2	234,0 V	235,7 V	0.72
3	235,0 V	236,0 V	0,42
4	233,6 V	235,1 V	0,64
5	234,5 V	236,6 V	0.89

Hasil perhitungan persentase *error* adalah sebagai berikut.

$$\text{Persentase } error = |(224,1-227,6)/224,1| \times 100\% = 3.9\%$$

$$\text{Persentase } error = |(234-235,7)/234| \times 100\% = 0.72\%$$

$$\text{Persentase } error = |(235-236)/235| \times 100\% = 0,42\%$$

$$\text{Persentase } error = |(233,6-235,1)/233,6| \times 100\% = 0,64\%$$

$$\text{Persentase } error = |(234,5-236,6)/234,5| \times 100\% = 0,89\%$$

- Pengujian Arus

Hasil pengujian arus ditunjukkan pada gambar 4.16 berikut.



Gambar 4.16 Pengujian Arus.

Hasil pengujian arus ditunjukkan pada tabel 4.3 berikut.

Tabel 4.3 Pengujian Arus

Pengujian	Beban	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase <i>Error %</i>
1	<i>Rice cooker</i>	1,64 A	1,62 A	1,21
2	Kipas Angin	0,124 A	0,12 A	3,22
3	Lampu	0,112 A	0,11 A	1,7
4	<i>Rice cooker</i>	1,65 A	1,61 A	2,42
5	<i>Rice cooker</i>	1,66 A	1,59 A	4,21

Hasil perhitungan persentase *error* adalah sebagai berikut.

$$\text{Persentase error} = |(1,64-1,62)/1,64| \times 100\% = 1,21\%$$

$$\text{Persentase error} = |(0,124 -0,12)/0,124| \times 100\% = 3,22\%$$

$$\text{Persentase error} = |(0,112 -0,11)/10,112| \times 100\% = 1,7\%$$

$$\text{Persentase error} = |(1,65 -1,61)/1,65| \times 100\% = 2,42\%$$

$$\text{Persentase error} = |(1,66-1,59)/1,66| \times 100\% = 4,21\%$$

- Pengujian Daya

Hasil pengujian daya ditunjukkan pada gambar 4.17 berikut.



Gambar 4.17 Pengujian Daya.

Hasil pengujian daya ditunjukkan pada tabel 4.4 berikut.

Tabel 4.4 Pengujian Daya

Pengujian	Beban	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase <i>Error</i> %
1	<i>Rice Cooker</i>	361,3 W	362 W	0,19
2	Kipas Angin	18 W	17,6 W	2,22
3	Solder	61 W	59,5 W	2,45
4	Lampu 1	12,3 W	12,1 W	1,62
5	Lampu 2	4,2 W	4,1 W	2,38

Hasil perhitungan persentase *error* adalah sebagai berikut.

$$\text{Persentase error} = |(361,3-362)/361,3| \times 100\% = 0,19\%$$

$$\text{Persentase error} = |(18 -17,6)/18| \times 100\% = 2,22\%$$

$$\text{Persentase error} = |(61 -59,5)/61| \times 100\% = 2,45\%$$

$$\text{Persentase error} = |(12,3 -12,1)/12,3| \times 100\% = 1,62\%$$

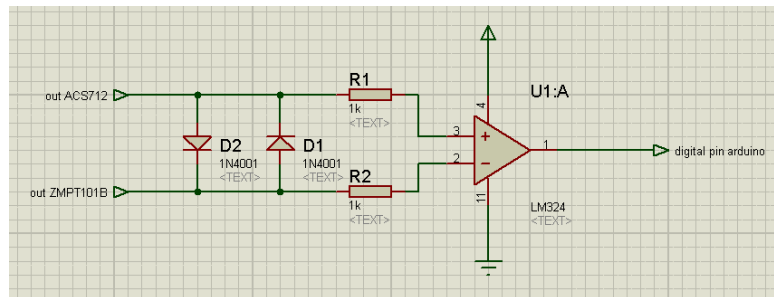
$$\text{Persentase error} = |(4,2-4)/4,2| \times 100\% = 2,38\%$$

4.3.5 Sensor Frekuensi

4.3.5.1 Perancangan Sensor Frekuensi

Pada penelitian ini digunakan modul sensor arus ACS712 dan modul sensor tegangan ZMPT101B sebagai pengkondisi sinyal frekuensi AC. Kedua sensor tersebut berfungsi sebagai pengganti transformator untuk menurunkan

tegangan AC 220V ke 5V tanpa merubah frekuensi yang sebenarnya. Lalu sinyal output kedua sensor tersebut di bandingkan menggunakan sebuah op-amp untuk merubah gelombang sinus menjadi sinyal kotak agar lebih mudah diolah oleh Arduino. Berikut adalah skematik dari rangkaian sensor frekuensi yang ditunjukkan pada gambar 4.18 dibawah ini.



Gambar 4.18 Skematik Rangkaian Sensor Frekuensi.

Adapun Sinyal keluaran dari op-amp ditunjukkan pada gambar 4.19 dibawah ini.

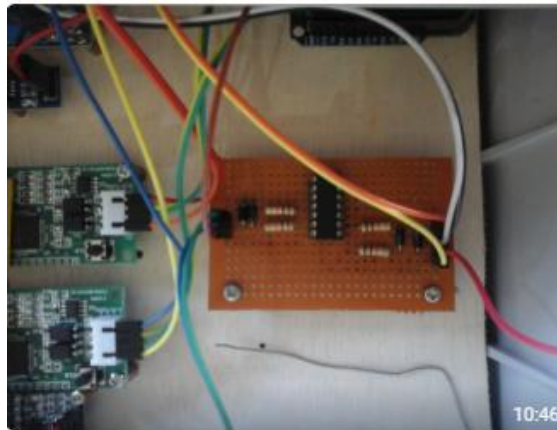


Gambar 4.19 Sinyal Keluaran Op-amp.

4.3.5.2 Pembuatan Sensor Frekuensi

Pembuatan sensor frekuensi ini dimulai dengan menyolder komponen op-amp pada papan pcb. Lalu merakit modul sensor ACS712 dan ZMPT101B sesuai dengan skematik yang telah dirancang sebelumnya. Setelah semua komponen siap, maka dilakukan perakitan seluruh komponen pada *box* panel. Adapun

penempatan komponen sensor frekuensi ditunjukkan pada gambar 4.20 dibawah ini.



Gambar 4.20 Penempatan Sensor Frekuensi

4.3.5.3 Pengujian Sensor Frekuensi

Pengujian sensor frekuensi dimulai dari pemasangan modul ACS712 dan ZMPT101B ke IC op-amp. *Output* analog modul sensor ACS712 dihubungkan ke pin *Non-inverting* op-amp, sedangkan *output* analog modul sensor ZMPT101B dihubungkan ke pin *Inverting* op-amp. Selanjutnya *output* dari op-amp dihubungkan ke pin *digital* Arduino. Pengujian dilakukan menggunakan pemrograman Arduino dengan *list* program sebagai berikut.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
int pin = 22;
double duration, duration1, f, t;

void setup()
{
  Serial.begin(9600);
  lcd.begin(20,4);
  lcd.backlight();
  pinMode(pin, INPUT);
}

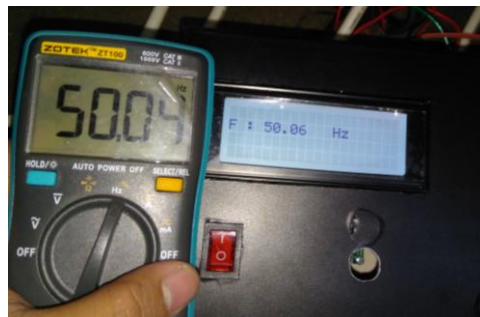
void loop()
{
  float rata=0,baca=0,sampling=0;
  unsigned long data = digitalRead(pin);
  if(data == 0){
    duration = pulseIn(pin, LOW);
```

```

duration = duration / 1000;
}
if(data == 1)
{
    duration1 = pulseIn(pin, HIGH);
    duration1 = duration1 / 1000;
}
t = ((duration + duration1)/1000);
if(t<=0){f=0;}
else {f = 1/t;}
lcd.setCursor(0,1);
lcd.print("F : ");
lcd.print(f);
lcd.setCursor(12,1);
lcd.print("Hz ");
delay(1000);
lcd.clear();
}

```

Adapun hasil pengujian sensor frekuensi dapat dilihat pada gambar 4.21 berikut.



Gambar 4.21 Pengujian Sensor Frekuensi.

Berikut adalah analisa pengujian sensor frekuensi yang ditunjukkan pada tabel 4.5.

Tabel 4.5 Pengujian Sensor Frekuensi

Pengujian	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase <i>Error %</i>
1	50,04 Hz	50,06 Hz	0,03
2	50,00 Hz	50,48 Hz	0,96
3	50,02 Hz	50,34 Hz	0,63
4	49,99 Hz	50,42 Hz	0,86
5	50,02 Hz	48,42 Hz	3,19

4.3.6.2 Pembuatan Sensor Suhu

Pembuatan sensor suhu dengan membuat sebuah *bracket* dudukan sensor pada body generator, lalu digunakan kabel *jumper* untuk menghubungkan sensor dengan pin Arduino. Pemasangan sensor suhu ditunjukkan pada gambar 4.23 dibawah ini.



Gambar 4.23 Pemasangan Sensor Suhu

4.3.6.3 Pengujian Sensor Suhu

Pengujian sensor suhu dimulai dari pemasangan pin VCC sensor ke catu daya +5V, pin GND sensor dihubungkan ke catu daya 0V, sedangkan pin output sensor dihubungkan ke pin *analog* a0 pada Arduino. Lalu sensor ditempelkan pada sebuah benda. Pengujian dilakukan menggunakan pemrograman Arduino dengan *list* program sebagai berikut.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
float tempC;
int tempPin= 3;

void setup() {
  Serial.begin(9600);
  lcd.begin(20,4);
  lcd.backlight();
}

void loop() {
  float rata=0,baca=0,sampling=0;
  for(int b=0;b<100;b++)
  {
    baca= analogRead(tempPin); // baca data dari sensor
    sampling=sampling+baca;
  }
}
```

```

rata =sampling/100;
tempC = ((5.0 * rata * 100.0)/1024.0*4.1)/5;
delay(3);
lcd.setCursor(0,1);
lcd.print("T : ");
lcd.print(tempC);
lcd.setCursor(12,1);
lcd.print("C ");
delay(1000);lcd.clear();
}

```

Adapun hasil pengujian sensor suhu dapat dilihat pada gambar 4.24 dan 4.25 dibawah ini.



Gambar 4.24 Pengujian Sensor Suhu.



Gambar 4.25 Tampilan Pengujian Sensor Suhu.

Berikut adalah analisa pengujian sensor frekuensi yang ditunjukkan pada tabel 4.6.

Tabel 4.6 Pengujian Sensor Suhu

Pengujian	Media	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase <i>Error %</i>
1	Setrika	43,5 °C	45,22 °C	3,95
2	Lantai	34,5 °C	34,87 °C	1,07
3	<i>Rice cooker</i>	56,7 °C	54,46 °C	3,95
4	Solder	52,4 °C	53,44 °C	1,98
5	generator	40,2 °C	41,55 °C	3,35

Hasil perhitungan persentase *error* adalah sebagai berikut.

$$\text{Persentase error} = |(43,5-45,22)/43,5| \times 100\% = 3,95\%$$

$$\text{Persentase error} = |(34,5-34,87)/34,5| \times 100\% = 1,07\%$$

$$\text{Persentase error} = |(56,7-54,46)/56,7| \times 100\% = 3,95\%$$

$$\text{Persentase error} = |(52,4-53,44)/52,4| \times 100\% = 1,98\%$$

$$\text{Persentase error} = |(40,2-41,55)/40,2| \times 100\% = 3,35\%$$

4.4 Sistem Interface

Software yang akan diterapkan pada sistem interface adalah *Visual Basic 2010 Express*. Pada Sistem *interface* ini, perangkat lunak dapat menampilkan nilai serta grafik hasil pengukuran dari seluruh sensor yang digunakan, meliputi nilai tegangan, arus, daya aktif, daya reaktif, daya semu, frekuensi, temperatur, dan faktor daya. Perangkat lunak ini di desain berkomunikasi dengan Arduino dalam pengiriman data secara *serial* melalui kabel *USB* yang terhubung langsung dari *port* data Arduino dengan *USB personal computer*.

4.4.1 Visual Basic

4.4.1.1 Perancangan Software Sistem Interface

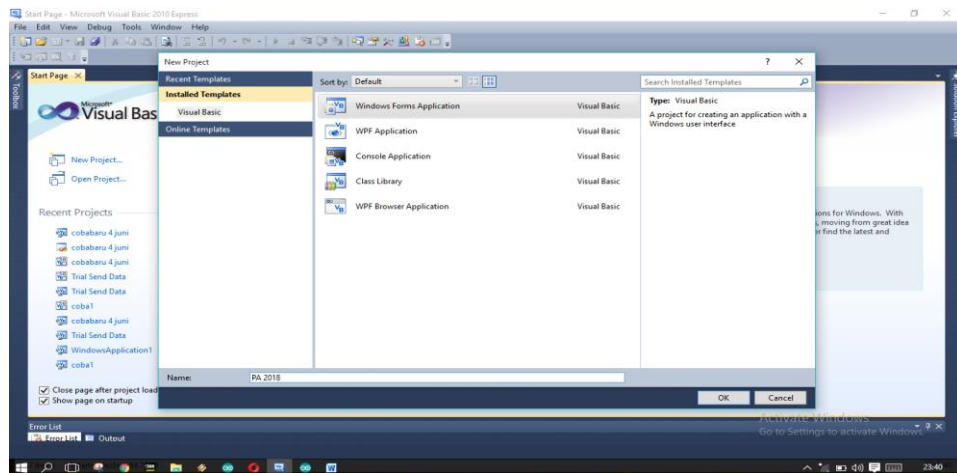
Perancangan software interface ini dilakukan pada *visual basic* dengan menggunakan *toolbox* yang tersedia di aplikasi tersebut untuk desain tampilan *interface*. *Toolbox* yang dipakai ada berbagai macam seperti *textbox*, *button*,

chart, label, grupbox, timer, serialport dan lainnya. Dan setelah didesain selanjutnya dibuat *list program* menggunakan bahasa *basic*.

4.4.1.2 Pembuatan Program *Software Sistem Interface*

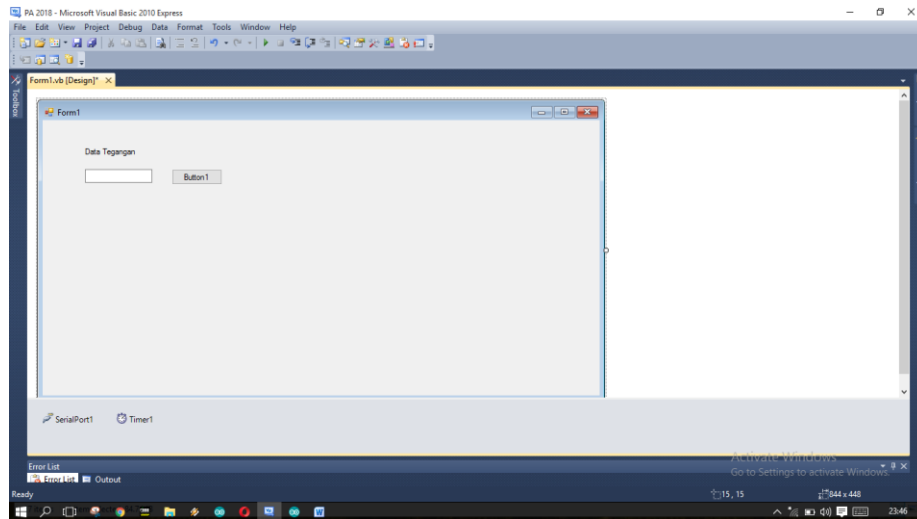
Pembuatan program *software interface* ini dimulai dengan mendesain tampilan *interface* dari *visual basic* dengan menggunakan *toolbox* dengan berbagai menu. Berikut adalah langkah-langkah pembuatan sistem *interface* menggunakan *software visual basic 2010 Express*

- Membuat *project* baru dengan cara klik *new project*, pilih Windows Form Application, kemudian ketikkan nama *projectnya* dan klik OK. Berikut adalah gambar cara membuat *project* baru pada *visual basic* ditunjukkan pada gambar 4.26 dibawah ini.



Gambar 4.26 Membuat *Project* Baru Pada *Visual Basic*.

- Membuat desain *layout* beserta komponen yang diperlukan pada bagian form1. Berikut adalah gambar *layout* form1 yang telah dibuat dapat dilihat pada gambar 4.27 dibawah ini.



Gambar 4.27 Desain *Layout* Form1.

Setelah itu pemrogramannya menggunakan bahasa *basic* yang saling berkaitan dengan arduino sehingga terjadi komunikasi *serial* yang baik. Berikut adalah *list program* form1 yang telah dibuat.

```
Imports System.IO
Imports System.IO.Ports
Imports System.Threading

Public Class Form1
    Shared _continue As Boolean
    Shared _serialPort As SerialPort
    Dim var1 As String
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SerialPort1.Close()
        SerialPort1.PortName = "com4" 'change com port to match your
Arduino port
        SerialPort1.BaudRate = 9600
        SerialPort1.DataBits = 8
        SerialPort1.Parity = Parity.None
        SerialPort1.StopBits = StopBits.One
        SerialPort1.Handshake = Handshake.None
        SerialPort1.Encoding = System.Text.Encoding.Default 'very
important!dat
        SerialPort1.Open()
        'Timer1.Enabled = True
        var1 = "0"

    End Sub

    Private Sub Button1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Timer1.Enabled = True
    End Sub
```

```

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    Timer1.Enabled = False
    SerialPort1.Write("1")
    TextBox1.Text = SerialPort1.ReadLine()
    Timer1.Enabled = True

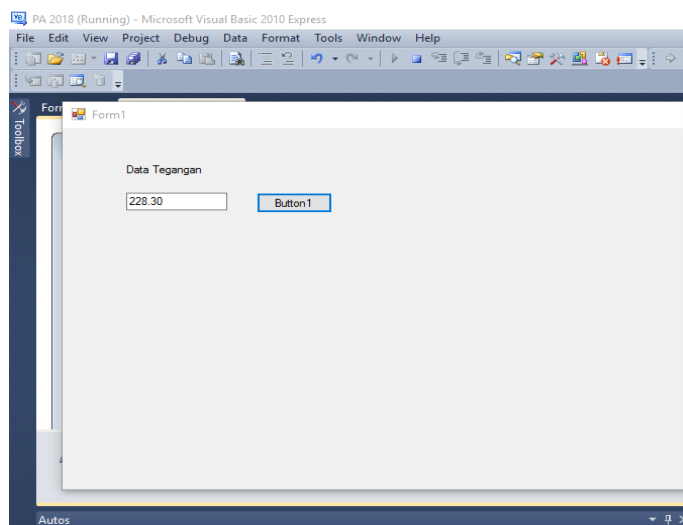
End Sub

End Class

```

4.4.1.3 Pengujian *Software Sistem Interface*

Pengujian dilakukan dengan mengirim data dari *visual basic* ke Arduino berupa *variabel*, selanjutnya dari arduino akan membaca variabel yang dikirim menggunakan menu *button*. Lalu setelah itu arduino akan mengirim data sensor ke *visual basic* yang akan ditampilkan menggunakan menu *textbox*. Berikut adalah hasil dari pengujian sistem *interface visual basic* dengan Arduino yang ditunjukkan pada gambar 4.28 dibawah ini.



Gambar 4.28 Gambar Hasil Pengujian Pembuatan Sistem *Interface*.

Berikut adalah *list program* Arduino yang dibuat untuk pengiriman data tegangan dari Arduino ke sistem *interface*.

```

#include <SoftwareSerial.h>
#include <PZEM004T.h>

PZEM004T pzem(&Serial1);
IPAddress ip(192,168,1,1);

```

```

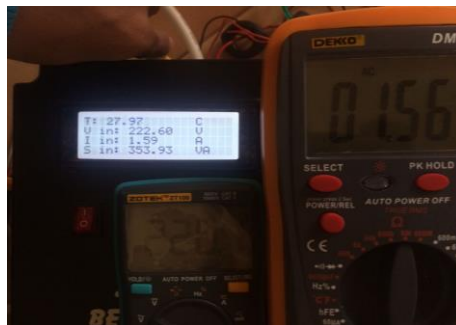
void setup() {
  Serial.begin(9600);
  pzem.setAddress(ip);
}

void loop() {
  int var= Serial.read()-'0';
  float v = pzem.voltage(ip);
  if (v < 0.0) v = 0.0;
  if (var==1){
    Serial.println(v);}
}

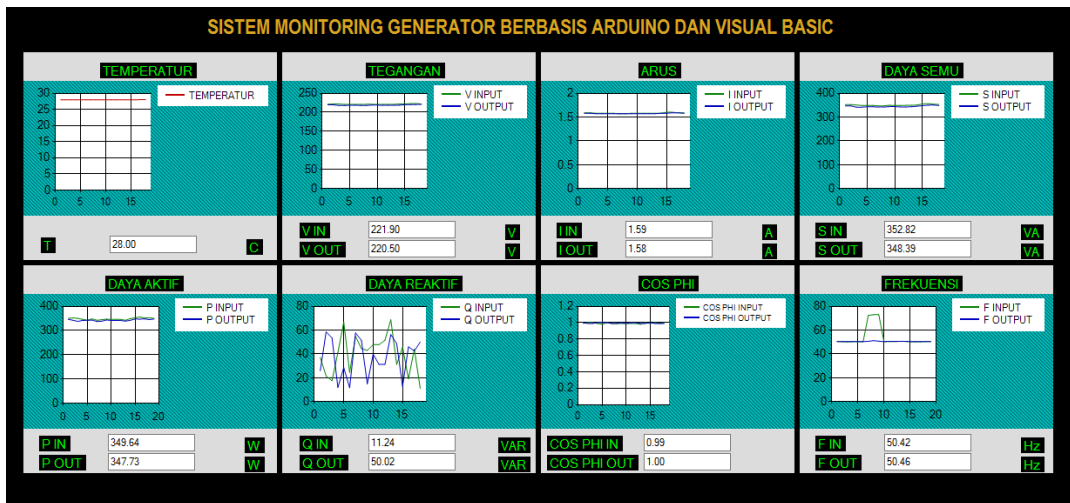
```

4.5 Pengujian Keseluruhan Alat

Pengujian keseluruhan alat meliputi pengujian blok *input* yang terdiri dari sensor daya, sensor frekuensi, dan sensor suhu, blok proses yang terdiri dari pengujian Arduino Mega 2560, sedangkan pada blok *output* meliputi pengujian tampilan LCD dan sistem *interface*. Pengujian ketiga blok ini dilakukan secara bersamaan. Berikut adalah hasil dari pengujian keseluruhan alat yang dicantumkan pada gambar dibawah ini. Pada gambar 4.29 dan gambar 4.30 merupakan pengukuran pada beban *rice cooker*.



Gambar 4.29 Hasil Pengujian Arus dan Tegangan Input.



Gambar 4.30 Hasil pengujian Sistem Interface.

Pada gambar 4.31, 4.32, 4.33 dan 4.34 merupakan hasil pengukuran terhadap beban kipas angin.



Gambar 4.31 Hasil Pengujian Daya dan Tegangan.



Gambar 4.32 Hasil Pengujian Arus dan Tegangan.



Gambar 4.33 Hasil Pengujian Daya Pada LCD.



Gambar 4.34 Hasil Pengujian Arus dan Tegangan Pada LCD.

Adapun analisa perhitungan pengujian ditunjukkan pada tabel 4.7.

Tabel 4.7 Hasil Pengujian Arus

Pengu jian	Beban	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase <i>Error</i> %	Frekue nsi (Hz)
1	Kipas Angin	0,128 A	0,12 A	6,25	50,47
2	<i>Rice Cooker</i>	1,56 A	1,59A	1,92	50,22
3	lampu	0,114 A	0,11 A	3,5	49,79
4	<i>Charger Laptop</i>	0,13 A	0,12 A	7,69	49,25
5	lampu	0,122 A	0,12 A	1,63	51,22

Hasil perhitungan persentase *error* adalah sebagai berikut.

$$\text{Persentase error} = |(0,128 - 0,12) / 0,128| \times 100\% = 6,25\%$$

$$\text{Persentase error} = |(1,56 - 1,59) / 1,56| \times 100\% = 1,92\%$$

$$\text{Persentase error} = |(0,114 - 0,11) / 0,114| \times 100\% = 3,5\%$$

$$\text{Persentase error} = |(0,13 - 0,12) / 0,13| \times 100\% = 7,96\%$$

$$\text{Persentase error} = |(0,122 - 0,12) / 0,122| \times 100\% = 1,63\%$$

Adapun analisa perhitungan pengujian ditunjukkan pada tabel 4.8.

Tabel 4.8 Hasil Pengujian Tegangan

Pengujian	Beban	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase Error %	Frekuensi (Hz)
1	Kipas Angin	234,9 V	231,1 V	1,61	50,47
2	<i>Rice Cooker</i>	235 V	237,4 V	1,02	50,22
3	lampu	234 V	236,6 V	1,11	49,79
4	<i>Charger Laptop</i>	233 V	235,6 V	1,11	49,25
5	lampu	231,1 V	234,6 V	1,51	51,22

Hasil perhitungan persentase *error* adalah sebagai berikut.

$$\text{Persentase error} = |(234,9 - 231,1) / 234,9| \times 100\% = 1,61\%$$

$$\text{Persentase error} = |(235 - 237,4) / 235| \times 100\% = 1,02\%$$

$$\text{Persentase error} = |(234 - 236,6) / 234| \times 100\% = 1,11\%$$

$$\text{Persentase error} = |(233 - 235,6) / 233| \times 100\% = 1,11\%$$

$$\text{Persentase error} = |(231,1 - 234,6) / 231,1| \times 100\% = 1,51\%$$

Adapun analisa perhitungan pengujian ditunjukkan pada tabel 4.9.

Tabel 4.9 Hasil Pengujian Daya

Pengujian	Beban	Alat Ukur Sebenarnya	Pengukuran Alat	Presentase Error (%)	Cos φ
1	Kipas Angin	18,4 W	18 W	2,17	0,76
2	<i>Rice Cooker</i>	362 W	361,8 W	0,05	0,92
3	lampu	4,1 W	3,92 W	4,36	0,21
4	<i>Charger Laptop</i>	15,9 W	15,3 W	3,7	0,54
5	lampu	4,1 W	4,00 W	2,4	0,20

Hasil perhitungan persentase *error* adalah sebagai berikut.

$$\text{Persentase } error = |(18,4 - 18) / 18,4| \times 100\% = 2,17\%$$

$$\text{Persentase } error = |(362 - 361,8) / 362| \times 100\% = 0,05\%$$

$$\text{Persentase } error = |(4,1 - 3,92) / 4,1| \times 100\% = 4,36\%$$

$$\text{Persentase } error = |(15,9 - 15,3) / 15,9| \times 100\% = 3,7\%$$

$$\text{Persentase } error = |(4,1 - 4,0) / 4,1| \times 100\% = 2,4\%$$

Kesimpulan yang didapatkan pada pengujian alat secara keseluruhan ini adalah sistem ini dapat mengukur tegangan pada generator dan beban dengan persentase *error* tertinggi pada beban kipas angin dengan error sebesar 1,61%. Pada pengujian arus, persentase *error* tertinggi didapat pada beban charger laptop dengan *error* sebesar 7,69%. Sedangkan pada pengujian daya, persentase *error* tertinggi didapat pada beban lampu dengan error sebesar 4,36%.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa terhadap fungsi alat pada Proyek Akhir dengan judul “Sistem Pemantauan Generator Berbasis Arduino dan *Visual Basic*” ini maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem Pemantauan Generator Berbasis Arduino dan *Visual basic* dapat memantau kinerja generator melalui *Personal computer (PC)* dan LCD. Hal ini memudahkan operator memantau kinerja generator.
2. Sistem Pemantauan Generator Berbasis Arduino dan *Visual basic* dapat menampilkan tegangan, arus, daya aktif, daya reaktif, daya semu, frekuensi dan temperatur dengan Arduino dengan toleransi kurang lebih 5%.

5.2 Saran

Apabila alat ini akan dikembangkan lebih lanjut, fungsi yang perlu diperbaiki dan ditambahkan antara lain :

1. Memperbaiki desain box Sistem Pemantauan Generator Berbasis Arduino dan *Visual basic* agar terlihat menarik.
2. Memperbaiki tampilan *interface* dari *visual basic* agar lebih menarik lagi dari apa yang telah kami buat sekarang.
3. Menguji sensor dengan tepat agar tidak terjadi kesalahan pada pengukuran yang dilakukan sensor.
4. Menggunakan dasar IOT agar dapat ditampilkan pada Android dan dengan jangkauan yang lebih luas.

DAFTAR PUSTAKA

- [1] WIKIPEDIA, Pembangkit listrik [Online], diakses pada tanggal 10 Mei 2018, Available : https://id.wikipedia.org/wiki/Pembangkit_listrik.
- [2] Alat Monitoring Pemakaian Energi Listrik Berbasis Android Menggunakan Modul PZEM-004T [Online], diakses pada tanggal 1 Agustus 2018, Available : sngbr.polinema.ac.id/index.php/sntet/article/download/81/77
- [3] Nyebarilmu.com, Cara mengakses sensor tegangan 220V ZMPT101B [Online], diakses pada tanggal 8 Agustus 2018, Available : <https://www.nyebarilmu.com/cara-mengakses-sensor-tegangan-220v-zmpt101b/>
- [4] ELECROW, ACS712 Current Sensor 5A [Online], diakses pada tanggal 8 Agustus 2018, Available : https://www.elecrow.com/wiki/index.php?title=ACS712_Current_Sensor_5A
- [5] Kl801, Penjelasan Tentang LM35 [Online], diakses pada 20 Juni 2017, Availbale : <http://kl801.ilearning.me/2015/05/21/penjelasan-tentang-lm35/>.
- [6] Codepolitan, Belajar IoT - Membuat Aplikasi Android untuk Memantau Suhu Ruangan Via Internet [Online], diakses pada tanggal 20 Mei 2018, Available : <https://www.codepolitan.com/belajar-iot-membuat-aplikasi-android-memantau-suhu-ruangan-via-internet>.
- [7] Arduino Info, Arduino Mega 2560 [Online], diakses pada tanggal 10 Mei 2018, Available : <https://arduino-info.wikispaces.com/MegaQuickRef>.
- [8] PENGENALAN VISUAL BASIC [Online], diakses pada tanggal 8 Agustus 2018, Available : margi.staff.gunadarma.ac.id/Downloads/files/43311/pengenalan-visual-basic-6-0.pdf

- [9] Webuilddatabases, Visual Basic Logo [Online], diakses pada tanggal 8 Agustus 2018, Available <https://webuilddatabases.com/technology/visual-basic-logo/>
- [10] Rapid, Winstar WH2004A-TMI-ET 20x4 LCD Display Blue Negative Mode White LED Backlight [Online], diakses pada tanggal 20 Mei 2018, Available : <https://www.rapidonline.com/winstar-wh2004a-tmi-et-20x4-lcd-display-blue-negative-mode-white-led-backlight-57-2254>.
- [11] Texas Instrument, PCF8574 Remote 8-bit I/O Expander for I2C bus [Online], diakses pada tanggal 1 Agustus 2018, Available : <http://www.tij.co.jp/product/jp/PCF8574/datasheet>.
- [12] Elektronika Dasar, LCD (Liquid Cristal Display) [Online], diakses pada 20 Mei 2018, Available : <http://elektronika-dasar.web.id/teori-elektronika/lcd-liquid-cristal-display/>.
- [13] picmicrolab, I2C Serial LCD Interface Module Arduino [Online], diakses pada tanggal 30 Juni 2018, Available : <http://www.picmicrolab.com/i2c-serial-lcd-interface-module-arduino>.
- [14] Kuntoro dan Monica Cinjy Saylendra, Makalah Tugas Akhir Polman Babel 2017, *MOBILE ROBOT DENGAN PENGONTROLAN VIA VOICE COMMAND* BERBASIS ANDROID.
- [15] MAKALAH FAKULTAS TEKNIK UNIVERSITAS INDONESIA 2010, *DAYA AKTIF, REAKTIF & NYATA* [Online] diakses pada tanggal 5 Agustus 2018, Available : <http://staff.ui.ac.id/system/files/users/chairul.hudaya/material/activeractiveandapparentpowerpaper.pdf>

LAMPIRAN

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Adam Zulhan
Tempat dan tanggal lahir : Palembang, 27 April 1998
Alamat rumah : Kp. Perumnas, Muntok, Bangka Barat
Telp : -
Hp : 0821 7757 4378
Email : adampolman0@gmail.com
Jenis kelamin : Laki-Laki
Agama : Islam



2. Riwayat Pendidikan

SD N 1	Muntok	2003-2009
SMP N 1	Muntok	2009-2012
SMK Bina Karya 1	Muntok	2012-2015

3. Pendidikan Non Formal

-

Sungailiat, 1 Agustus 2018



Adam Zulhan

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Aji Kusuma
Tempat dan tanggal lahir : Muntok, 4 Juni 1997
Alamat rumah : Kp. Senang Hati, Muntok,
Bangka Barat
Telp : -
Hp : 0856 0967 3197
Email : ajikusumapolman@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



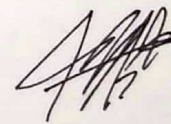
2. Riwayat Pendidikan

SD N 7	Muntok	2003-2009
SMP N 1	Muntok	2009-2012
SMK N 1	Muntok	2012-2015

3. Pendidikan Non Formal

-

Sungailiat, 1 Agustus 2018



Aji Kusuma

LAMPIRAN 2

**Program Arduino Mega 2560: Sistem Pemantauan Generator
Berbasis Arduino dan Visual Basic**


```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

PZEM004T pzem(&Serial1); // (RX,TX) connect to TX,RX of PZEM
PZEM004T pzem1(&Serial2); // (RX,TX) connect to TX,RX of PZEM
IPAddress ip(192,168,1,1);

////////////////////////////////////

int pin = 23;
int pin1 = 22;
double duration, duration1, f, t;
double durasi, durasi1, f1, t1;

float v,v1,i,i1,s,s1,cosphi,cosphi1;
float p,p1,q,q1;
////////////////////////////////////

float tempC;
int tempPin= 3;
int a;

////////////////////////////////////FUNGSI SUHU////////////////////////////////////

void suhu()
{
float rata=0,baca=0,sampling=0;
for(int b=0;b<100;b++)

```

```

{
  baca= analogRead(tempPin);// baca data dari sensor
  sampling=sampling+baca;
}
rata =sampling/100;
tempC = ((5.0 * rata * 100.0)/1024.0*4.1)/5;
delay(3);
}

```

//////////////////////////////////FUNGSI FREKUENSI 1//////////////////////////////////

```

void frekuensi()
{
  unsigned long data = digitalRead(pin);
  if(data == 0)
  {
    duration = pulseIn(pin, LOW);
    duration = duration / 1000;
  }
  if(data == 1)
  {
    duration1 = pulseIn(pin, HIGH);
    duration1 = duration1 / 1000;
  }
  t = ((duration + duration1)/1000);
  if(t<=0||v<=0){f=0;}
  else {f = 1/t;}
}

```

//////////////////////////////////FUNGSI FREKUENSI 2//////////////////////////////////

```

void frekuensi1()

```

```

{
  unsigned long nilai = digitalRead(pin1);
  if(nilai == 0)
  {
    durasi = pulseIn(pin1, LOW);
    durasi = durasi / 1000;
  }
  if(nilai == 1)
  {
    durasi1 = pulseIn(pin1, HIGH);
    durasi1 = durasi1 / 1000;
  }
  t1 = ((durasi + durasi1)/1000);
  if(t1<=0||v1<=0){f1=0;}
  else {f1 = 1/t1;}

}

```

////////PZEM SENSOR ////////////

```

void hasil()
{
  v = pzem.voltage(ip);
  v1 = pzem1.voltage(ip);
  i = pzem.current(ip);
  i1 = pzem1.current(ip);
  p = pzem.power(ip);
  p1 = pzem1.power(ip);

  if (v < 0.0){v=0.0;}
  if (i < 0.0){i=0.0;}
  if (p < 0.0){p=0.0;}

```

```

if (v1 < 0.0){v1=0.0;}
if (i1 < 0.0){i1=0.0;}
if (p1 < 0.0){p1=0.0;}
cosphi=p/(i*v);
s=i*v;
if (cosphi>1){cosphi=1-(cosphi-1.00);}
p=s*cosphi;
float se=s*s;
float pa=p*p;
float la=se-pa;
q=sqrt(la);

cosphi1 = p1/(i1*v1);
s1 = i1*v1;
if (cosphi1>1){cosphi1=1-(cosphi1-1.00);}
p1=s1*cosphi1;
float se1=s1*s1;
float pa1=p1*p1;
float la1=se1-pa1;

q1=sqrt(la1);
}

```

```

void setup()
{
  Serial.begin(9600);
  pzem.setAddress(ip);
  pzem1.setAddress(ip);

  a=0;
  pinMode(pin, INPUT);

```

```

pinMode(pin1, INPUT);
lcd.begin(20,4);
lcd.backlight();

/////OPENING DISPLAY//////////

lcd.setCursor(0,0);
lcd.print(" SISTEM PEMANTAUAN ");

lcd.setCursor(0,1);
lcd.print(" GENERATOR BERBASIS");

lcd.setCursor(0,2);
lcd.print(" ARDUINO DAN VISUAL");

lcd.setCursor(0,3);
lcd.print(" BASIC");
delay(2500);
}

void loop()
{
  suhu(); //Fungsi sensor Suhu
  hasil(); //Fungsi sensor pzem 1
  frekuensi(); //Fungsi sensor Frekuensi 1
  frekuensi1(); //Fungsi sensor Frekuensi 2
  a++;
  int val = Serial.read() - '0';

  ///////////SERIAL COMMUNICATION//////////
  if (val == 1)
  {

```

```
Serial.println(tempC);  
Serial.println(v);  
Serial.println(i);  
Serial.println(s);  
Serial.println(p);  
Serial.println(q);  
Serial.println(cosphi);  
Serial.println(f);  
Serial.println(v1);  
Serial.println(i1);  
Serial.println(s1);  
Serial.println(p1);  
Serial.println(q1);  
Serial.println(cosphi1);  
Serial.println(f1);  
Serial.println(tempC);  
Serial.println(v);  
Serial.println(i);  
Serial.println(s);  
Serial.println(p);  
Serial.println(q);  
Serial.println(cosphi);  
Serial.println(f);  
Serial.println(v1);  
Serial.println(i1);  
Serial.println(s1);  
Serial.println(p1);  
Serial.println(q1);  
Serial.println(cosphi1);  
Serial.println(f1);  
Serial.flush();
```

```
}
```

```
//////////LCD DISPLAY PERTAMA//////////
```

```
if(a<=3)
```

```
{
```

```
  lcd.clear();
```

```
  lcd.setCursor(0,0);
```

```
  lcd.print("T: ");
```

```
  lcd.print(tempC);
```

```
  lcd.setCursor(15,0);
```

```
  lcd.print("C ");
```

```
  lcd.setCursor(0,1);
```

```
  lcd.print("V in: ");
```

```
  lcd.print(v);
```

```
  lcd.setCursor(14,1);
```

```
  lcd.print(" V");
```

```
  lcd.setCursor(0,2);
```

```
  lcd.print("I in: ");
```

```
  lcd.print(i);
```

```
  lcd.setCursor(14,2);
```

```
  lcd.print(" A");
```

```
  lcd.setCursor(0,3);
```

```
  lcd.print("S in: ");
```

```
  lcd.print(s);
```

```
  lcd.setCursor(14,3);
```

```
  lcd.print(" VA");
```

```
}
```

```
//////////LCD DISPLAY KEDUA//////////
```

```
if(a>=4&&a<=6)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("P in: ");
  lcd.print(p);
  lcd.setCursor(15,0);
  lcd.print("W ");

  lcd.setCursor(0,1);
  lcd.print("Q in: ");
  lcd.print(q);
  lcd.setCursor(14,1);
  lcd.print(" VAR");

  lcd.setCursor(0,2);
  lcd.print("Cosphi in: ");
  lcd.print(cosphi);

  lcd.setCursor(0,3);
  lcd.print("f in: ");
  lcd.print(f);
  lcd.setCursor(15,3);
  lcd.print("Hz ");
}
```

```
//////////LCD DISPLAY KETIGA//////////
```

```
if(a>=7&&a<=9)
{
```



```
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("T: ");  
lcd.print(tempC);  
lcd.setCursor(15,0);  
lcd.print("C ");
```

```
lcd.setCursor(0,1);  
lcd.print("Vout: ");  
lcd.print(v1);  
lcd.setCursor(14,1);  
lcd.print(" V");
```

```
lcd.setCursor(0,2);  
lcd.print("Iout: ");  
lcd.print(i1);  
lcd.setCursor(14,3);  
lcd.print("A ");
```

```
lcd.setCursor(0,3);  
lcd.print("Sout: ");  
lcd.print(s1);  
lcd.setCursor(14,3);  
lcd.print("VA ");  
}
```

```
//////////LCD DISPLAY KEEMPAT//////////
```

```
if(a>=10&&a<=12)  
{  
lcd.clear();  
lcd.setCursor(0,0);
```

```
lcd.print("Pout: ");  
lcd.print(p1);  
lcd.setCursor(15,0);  
lcd.print("W ");
```

```
lcd.setCursor(0,1);  
lcd.print("Qout: ");  
lcd.print(q1);  
lcd.setCursor(14,1);  
lcd.print(" VAR");
```

```
lcd.setCursor(0,2);  
lcd.print("Cosphi out: ");  
lcd.print(cosphi1);
```

```
lcd.setCursor(0,3);  
lcd.print("fout: ");  
lcd.print(f1);  
lcd.setCursor(15,3);  
lcd.print("Hz ");  
}
```

```
if(a>=13)  
{  
  a=0;  
}  
}
```

LAMPIRAN 3

**Program *Visual Basic* : Sistem Pemantauan Generator Berbasis
*Arduino dan Visual Basic***

-FORM 1

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If TextBox1.Text = "admin" And TextBox2.Text = "polman123" Then
            MsgBox("MASUK KE SISTEM")
            Form4.Show()
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button2.Click
        TextBox1.Clear()
        TextBox2.Clear()
        TextBox1.Focus()
    End Sub

End Class
```

-FORM 2

```
Imports System.IO
Imports System.IO.Ports
Imports System.Threading

Public Class Form2
    Shared _continue As Boolean
    Shared _serialPort As SerialPort
    Dim var1 As String
    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        SerialPort1.Close()
        SerialPort1.PortName = "com4" 'change com port to match your
        Arduino port
        SerialPort1.BaudRate = 9600
        SerialPort1.DataBits = 8
        SerialPort1.Parity = Parity.None
        SerialPort1.StopBits = StopBits.One
        SerialPort1.Handshake = Handshake.None
        SerialPort1.Encoding = System.Text.Encoding.Default 'very
        important!dat
        SerialPort1.Open()
        'Timer1.Enabled = True
        var1 = "0"

    End Sub
```

```

    Private Sub Button1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Timer1.Enabled = True
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Timer1.Enabled = False
        SerialPort1.Write("1")
        TextBox1.Text = SerialPort1.ReadLine()
        TextBox2.Text = SerialPort1.ReadLine()
        TextBox3.Text = SerialPort1.ReadLine()
        TextBox4.Text = SerialPort1.ReadLine()
        TextBox5.Text = SerialPort1.ReadLine()
        TextBox6.Text = SerialPort1.ReadLine()
        TextBox7.Text = SerialPort1.ReadLine()
        TextBox8.Text = SerialPort1.ReadLine()
        TextBox9.Text = SerialPort1.ReadLine()
        TextBox10.Text = SerialPort1.ReadLine()
        TextBox11.Text = SerialPort1.ReadLine()
        TextBox12.Text = SerialPort1.ReadLine()
        TextBox13.Text = SerialPort1.ReadLine()
        TextBox14.Text = SerialPort1.ReadLine()
        TextBox15.Text = SerialPort1.ReadLine()
        Me.Chart1.Series("TEMPERATUR").Points.AddY(SerialPort1.ReadLine())
        Me.Chart2.Series("V INPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart3.Series("I INPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart4.Series("S INPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart5.Series("P INPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart8.Series("Q INPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart7.Series("COS PHI
INPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart6.Series("F INPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart2.Series("V OUTPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart3.Series("I OUTPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart4.Series("S OUTPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart5.Series("P OUTPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart8.Series("Q OUTPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart7.Series("COS PHI
OUTPUT").Points.AddY(SerialPort1.ReadLine())
        Me.Chart6.Series("F OUTPUT").Points.AddY(SerialPort1.ReadLine())
        Timer1.Enabled = True

    End Sub

    Private Sub Label21_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label21.Click

    End Sub
End Class

```

LAMPIRAN 4

Desain *Layout Visual Basic* : Sistem Pemantauan Generator

Berbasis *Arduino* dan *Visual Basic*

-Desain FORM 1

The image shows a screenshot of a web application window titled "Form1". The window has a light blue header bar with the text "SELAMAT DATANG PADA SISTEM MONITORING GENERATOR" in dark blue. Below the header, there is a large graphic consisting of a grey downward-pointing chevron and a dark blue upward-pointing chevron. In the center, there is a teal-colored login form. The form contains the text "Please Login..." followed by two input fields labeled "Username" and "Password". Below the input fields are two buttons: "LOGIN" and "CLEAR". At the bottom of the window, the text "POLMAN BABEL" is displayed in a large, blue, sans-serif font. The window also features standard Windows-style window controls (minimize, maximize, close) in the top right corner.

-Desain FORM 2

Form2

SISTEM MONITORING GENERATOR BERBASIS ARDUINO DAN VISUAL BASIC

TEMPERATUR	TEGANGAN	ARUS	DAYA SEMU
T <input type="text"/> C	V IN <input type="text"/> V V OUT <input type="text"/> V	I IN <input type="text"/> A I OUT <input type="text"/> A	S IN <input type="text"/> VA S OUT <input type="text"/> VA
DAYA AKTIF	DAYA REAKTIF	COS PHI	FREKUENSI
P IN <input type="text"/> W P OUT <input type="text"/> W	Q IN <input type="text"/> VAR Q OUT <input type="text"/> VAR	COS PHI IN <input type="text"/> COS PHI OUT <input type="text"/>	F IN <input type="text"/> Hz F OUT <input type="text"/> Hz

[Baca](#)

Activate Windows
Go to Settings to activate