

**SISTEM MONITORING 3 PEMBANGKIT HIBRIDA PADA
LABORATORIUM SISTEM KETENAGAAN UNIVERSITAS
ISLAM INDONESIA**

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



Disusun oleh:

Surya Hari Arafah

15524024

**Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta**

2021

LEMBAR PENGESAHAN

JUDUL SKRIPSI UNTUK S1 TEKNIK ELEKTRO UII

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia**

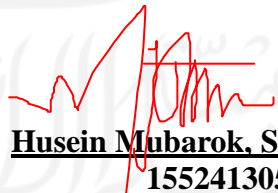
Disusun oleh:

**Surya Hari Arafah
15524024**

Yogyakarta, 2 Oktober 2021

Menyetujui,

Pembimbing 1


**Husein Mubarak, S.T., M.Eng
155241305**

LEMBAR PENGESAHAN

SKRIPSI

JUDUL SKRIPSI UNTUK S1 TEKNIK ELEKTRO UII

Dipersiapkan dan disusun oleh:

SURYA HARI ARAFAH

15524024

Telah dipertahankan di depan dewan penguji

Pada tanggal: 26 November 2021

Susunan dewan penguji

Ketua Penguji : Husein Mubarak, S.T., M.Eng,

Anggota Penguji 1: Firdaus, S.T., M.T., Ph.D,

Anggota Penguji 2: Medilla Kusriyanto, S.T., M.Eng,

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Tanggal: 26 November 2021

Ketua Program Studi Teknik Elektro



Yusuf Aziz Amrullah S.T., M.Eng., Ph.D.

045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 6 Juni 2021



Surya Hari Arafah

KATA PENGANTAR

Segala puji bagi Allah tuhan semesta alam yang telah melimpahkan rahmat, hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan laporan skripsi ini dengan judul “Desain Sistem SCADA untuk Monitoring 3 Pembangkit Hibrida pada Laboratorium Sistem Ketenagaan Universitas Islam Indonesia” sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia. Tidak lupa shalawat serta salam semoga tetap tercurah limpahkan kepada manusia mulia yang pernah hidup di muka bumi ini yakni nabi Muhammad shallaahu a’laihi wasallam beserta keluarga, sahabat, dan para pengikutnya yang senantiasa mengikuti sunnah-sunnahnya hingga akhir zaman nanti. Semoga kita mendapatkan syafa’atnya ketika hari akhir kelak tiba.

Tentunya penulis menyadari bahwa skripsi ini tidak akan terselesaikan tanpa adanya do’a, usaha, kerja keras serta dukungan dari berbagai pihak terkait. Maka dengan kerendahan hati dan setulus-tulusnya ucapan penulis ingin menyampaikan rasa terimakasih yang sebesar-besarnya kepada:

1. Kedua orang tua saya yakni abah Syahrizal dan umi Maliyar Dewi yang selalu memberikan semangat serta mendo’akan penulis di sela-sela sujudnya.
2. Abang saya Surya Ilham Syahputra yang telah membiayai penulis semasa di bangku perkuliahan.
3. Bapak Yusuf Aziz Amrulloh S.T., M.Eng., Ph.D. selaku kepala prodi jurusan Teknik Elektro Universitas Islam Indonesia.
4. Bapak Husein Mubarak S.T., M.Eng selaku Pembimbing penulis yang telah meluangkan waktunya untuk membantu, membimbing penulis dengan penuh kesabaran.
5. Segenap dosen jurusan Teknik Elektro Universitas Islam Indonesia yang telah memberikan ilmunya semasa di bangku perkuliahan. Semoga menjadi amal jariyah untuk Bapak / Ibu sekalian.
6. Mas Hendri selaku laboran Laboratorium Ketenagaan Teknik Elektro Universitas Islam Indonesia yang sudah membantu menyediakan segala alat dan bahan selama pengambilan data Skripsi.
7. Seluruh teman-teman, keluarga besar Teknik Elektro Universitas Islam Indonesia dan pihak-pihak terkait dalam penulisan Skripsi yang tidak dapat saya tuliskan namanya satu persatu.
8. Semua orang yang menanyakan kapan saya lulus.

Semoga Allah membalas seluruh kebaikan Bapak/Ibu/Saudara sekalian serta mempermudah segala urusan baik selama di dunia maupun di akhirat kelak. Penulis menyadari bahwa pasti ada kekurangan di dalam penulisan Skripsi ini karena keterbatasan ilmu pengetahuan yang dimiliki. Maka dari itu penulis mengharapkan kritik dan saran yang membangun sehingga dapat melengkapi kekurangan-kekurangan yang ada. Semoga Skripsi ini dapat memberi manfa'at bagi para pembacanya. Aamiin.



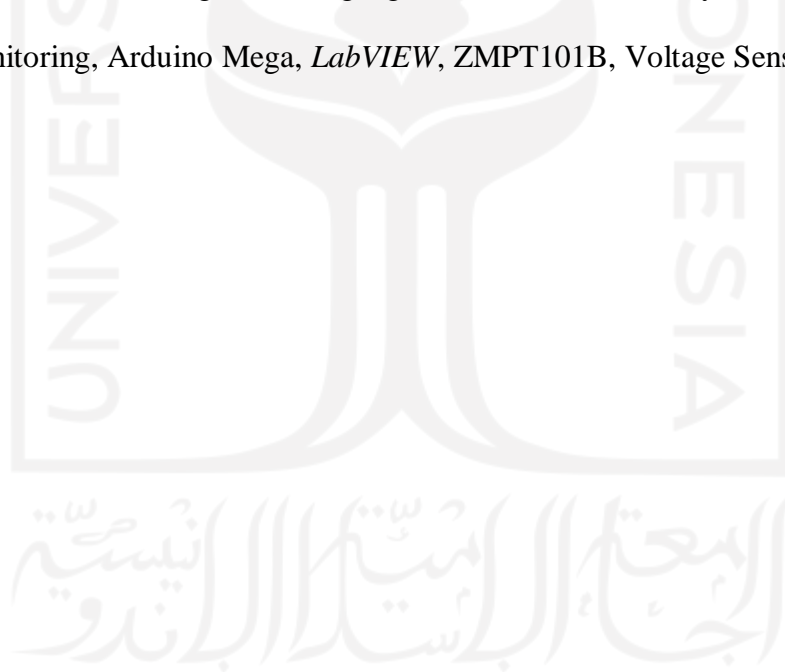
ARTI LAMBANG DAN SINGKATAN

IPTEK	Ilmu Pengetahuan dan Teknologi
IC	<i>Integrated Circuit</i>
LabVIEW	<i>Laboratory Virtual Instrumentation Engineering Workbench</i>
GSM	<i>Global System for Mobile Communication</i>
PV	<i>Photovoltaic</i>
SMS	<i>Short Message Service</i>
GUI	<i>Graphics User Interface</i>
w	watt
WP	Watt Peak
PC	<i>Personal Computer</i>
RAM	<i>Random Acces Memory</i>
ROM	<i>Read Only Memory</i>
CPU	<i>Central Processing Unit</i>
EPROM	<i>Erasable Programmable Read Only Memory</i>
EEPROM	<i>Electrically Erasable Programmable Read Only Memory</i>
LCD	<i>Liquid Crystal Display</i>
I/O	<i>Input / Output</i>
DAC	<i>Digital to Analog Converter</i>
ADC	<i>Analog to Digital Converter</i>
VI	<i>Virtual Instrumens</i>
PLTMH	Pembangkit Listrik Tenaga Mikrohidro
PLTA	Pembangkit Listrik Tenaga Air
kw	kilo watt
GGL	Gaya Gerak Listrik
P	Daya
V	Tegangan
I	Arus
PLTBS	Pembangkit Listrik Tenaga Bayu
h	hours
Ah	Ampere hours
SCC	<i>Solar Cell Controller</i>
f / Hz	frekuensi / Hertz
R / Ω	resistansi / Ohm
L / H	induktansi / henry
C / F	kapasitansi / farad
AC / DC	<i>Alternating Current / Direct Current</i>
PLN	Perusahaan Listrik Negara
Vrms	<i>Volt root mean square</i>
Irms	<i>Ampere root mean square</i>
USB	<i>Universal Serial Bus</i>
LAN	<i>Local Area Network</i>
cos ϕ	faktor daya
PWM	<i>Pulse Width Modulation</i>
SRAM	<i>Static Random Acces Memory</i>
IoT	<i>Internet of Things</i>

ABSTRAK

Menginjak era revolusi industri 4.0, perkembangan teknologi cukup pesat terutama dalam bidang otomasi industri. Salah satu aplikasi yang digunakan dalam proses otomasi yakni aplikasi sistem monitoring. Sistem monitoring berfungsi untuk memantau data yang dihasilkan dari suatu ataupun beberapa unit beban di sebuah *plant* secara *real time* agar data tersebut nantinya dapat dianalisa. Dengan memanfaatkan arduino mega dan *LabVIEW* diharapkan dapat menampilkan data keluaran dari suatu *plant* secara *real time* tanpa harus kontak langsung untuk mengukur dan mencatat hasil *output* sebuah *plant* secara manual. Beberapa alat yang digunakan yakni sensor *zmpt101b* untuk mengukur tegangan *AC*, voltage sensor untuk mengukur tegangan *DC*, *acs712* untuk mengukur arus *AC* maupun arus *DC*. Arduino mega berfungsi untuk mengolah data yang dibaca oleh sensor *zmpt101b*, voltage sensor dan *acs712*. *LabVIEW* berfungsi untuk menampilkan data yang diolah oleh arduino mega baik dalam bentuk grafik ataupun angka. Dari hasil penelitian yang dilakukan didapatkan persentase keberhasilan alat yang dibuat dengan rincian alat yakni *zmpt101b* 98,33%, voltage sensor 99,94%, *acs712* arus *AC* 92,19% dan *acs712* arus *DC* 92,94%. Persentase rata-rata *error* data antara yang tertampil di serial monitor dengan *LabVIEW* pada pengujian *PLTMH*, *Solar Cell* dan *PLTB* sudah termasuk golongan alat SNI dimana nilai yang didapatkan dibawah 5%. Pembacaan data pada front panel *LabVIEW* juga sudah cukup jelas dan mudah dipahami dimana kolom pembacaan output dari *PLTMH* dan *PLTB* berada di posisi bagian sebelah kiri dengan nilai tegangan *AC* dan arus *AC*, sedangkan output dari *solar cell* berada di posisi bagian sebelah kanan dengan nilai tegangan *DC*, arus *DC* dan daya *DC*.

Kata kunci: Monitoring, Arduino Mega, *LabVIEW*, *ZMPT101B*, Voltage Sensor, *ACS712*.



DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PENGESAHAN	i
PERNYATAAN	ii
KATA PENGANTAR	iii
ARTI LAMBANG DAN SINGKATAN	v
ABSTRAK	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
BAB 2 TINJAUAN PUSTAKA	1
2.1 Studi Literatur	1
2.2 Tinjauan Teori	2
2.2.1 Sistem Monitoring	2
2.2.2 Mikrokontroler	3
2.2.3 Arduino IDE	5
2.2.4 <i>LabVIEW</i>	5
BAB 3 METODOLOGI	8
3.1 Alat dan Bahan	8
3.2 Alur Penelitian	8

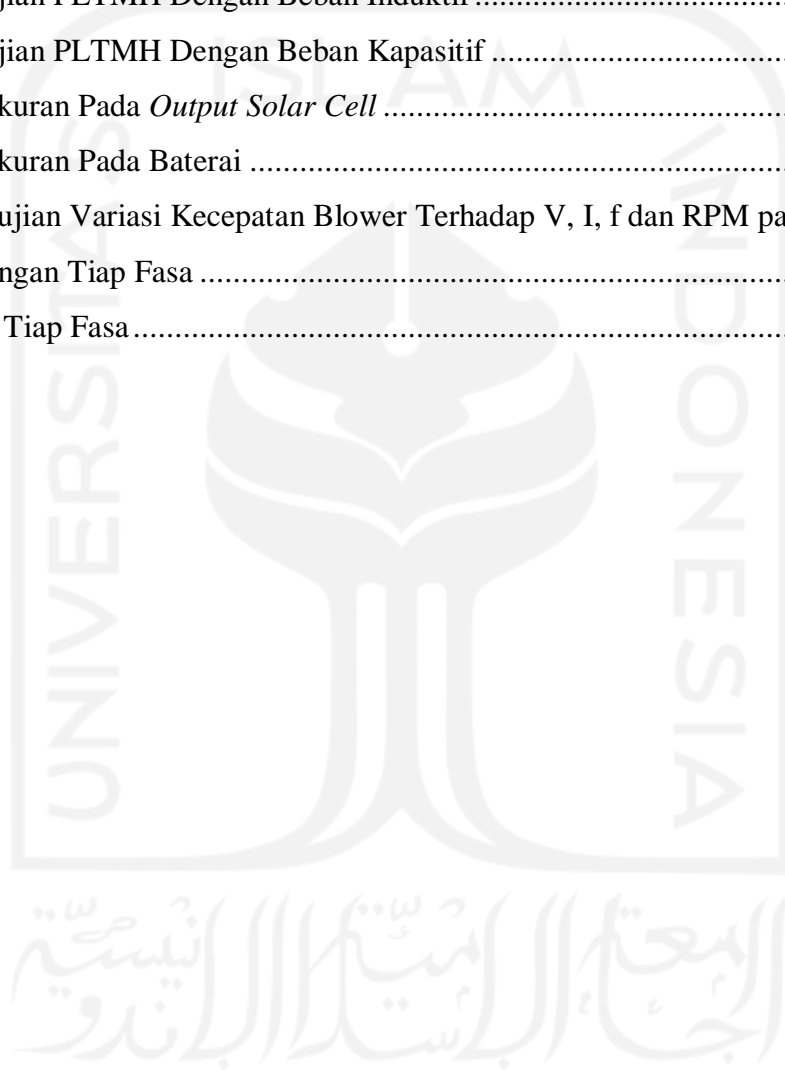
3.3 Proses Pengiriman Data Sistem Monitoring	10
3.3.1 Sensor.....	11
3.3.2 Arduino Mega.....	13
3.3.3 Laptop	13
3.4 Rangkaian Pengujian Sistem Monitoring	15
3.5 Cara Analisis	16
BAB 4 HASIL DAN PEMBAHASAN	18
4.1 Karakteristik Sensor ZMPT101B	18
4.2 Karakteristik Voltage Sensor	18
4.3 Karakteristik Sensor ACS712 Arus AC.....	19
4.4 Karakteristik Sensor ACS712 Arus DC.....	19
4.5 Pengujian Beban RLC Terhadap V, I, f dan $\cos\phi$ Pada PLTMH.....	20
4.5.1 Data Pengujian PLTMH Dengan Beban Resistif	20
4.5.2 Data Pengujian PLTMH Dengan Beban Induktif	20
4.5.3 Data Pengujian PLTMH Dengan Beban Kapasitif.....	21
4.6 Pengujian <i>Solar Cell</i>	22
4.7 Pengujian Variasi Kecepatan Blower Terhadap V, I, f dan RPM pada PLTB	23
4.8 Penilaian <i>User</i> Terhadap Tampilan dan Fungsi Sistem Monitoring	26
BAB 5 KESIMPULAN DAN SARAN	28
5.1 Kesimpulan	28
5.2 Saran	28
DAFTAR PUSTAKA.....	29
LAMPIRAN.....	1

DAFTAR GAMBAR

Gambar 2.1 Mikrokontroler	3
Gambar 2.2 Diagram Mikrokontroler.....	3
Gambar 2.3 Arduino IDE.....	5
Gambar 2.4 Front Panel.....	6
Gambar 2.5 Block Diagram	6
Gambar 2.6 Control Pallete.....	7
Gambar 2.7 Functions Pallete	7
Gambar 3.1 Flowchart Penelitian.....	9
Gambar 3.2 Diagram Blog Penelitian	11
Gambar 3.3 Voltage Sensor	12
Gambar 3.4 Sensor ZMPT101B.....	12
Gambar 3.5 Sensor ACS712 dan Skematik ACS712	13
Gambar 3.6 Arduino Mega	13
Gambar 3.7 Tampilan Monitoring Pada Block Diagram <i>LabVIEW</i>	14
Gambar 3.8 Case Structure	14
Gambar 3.9 Rangkaian Pengujian PLTMH	15
Gambar 3.10 Rangkaian Pengujian <i>Solar Cell</i>	15
Gambar 3.11 Rangkaian Pengujian PLTB.....	16
Gambar 3.12 Hubungan Bintang.....	17
Gambar 4.1 Tampilan Monitoring Front Panel Pada <i>LabVIEW</i>	26
Gambar 4.2 Penilaian User Terhadap Tampilan dan Fungsi Sistem Monitoring	27

DAFTAR TABEL

Tabel 4.1 Karakteristik Sensor ZMPT101B	18
Tabel 4.2 Karakteristik Voltage Sensor.....	18
Tabel 4.3 Karakteristik Sensor ACS712 Arus AC	19
Tabel 4.4 Karakteristik Sensor ACS712 Arus DC	19
Tabel 4.5 Pengujian PLTMH Dengan Beban Resistif.....	20
Tabel 4.6 Pengujian PLTMH Dengan Beban Induktif	21
Tabel 4.7 Pengujian PLTMH Dengan Beban Kapasitif	21
Tabel 4.8 Pengukuran Pada <i>Output Solar Cell</i>	22
Tabel 4.9 Pengukuran Pada Baterai	22
Tabel 4.10 Pengujian Variasi Kecepatan Blower Terhadap V, I, f dan RPM pada PLTB	24
Tabel 4.11 Tegangan Tiap Fasa	25
Tabel 4.12 Arus Tiap Fasa	25



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Begitu pesatnya pertumbuhan ilmu pengetahuan dan teknologi (IPTEK) di dunia industri mengharuskan perusahaan mengambil alih dalam perkembangannya bagi setiap negara termasuk Indonesia. Menurut menteri perindustrian Indonesia yakni bapak Airlangga Hartanto mengatakan bahwa Indonesia masuk ke dalam 5 besar tertinggi dalam sumbangsuhnya terhadap dunia perindustrian di dunia. “Capaian 20 persen sangatlah besar, sehingga Indonesia masuk dalam jajaran elit dunia. Dalam kategori *manufacturing value added*, Indonesia masuk dalam 10 besar dunia. Peringkat ini sejajar dengan Brazil dan Inggris serta lebih besar dari Rusia,” kata Menperin ketika menyampaikan orasi ilmiah pada Wisuda Universitas Muhammadiyah Malang ke 86 Periode IV Tahun 2017 di Malang, Sabtu (25/11).

Menginjak era revolusi industri 4.0, perkembangan teknologinya cukup pesat. Tidak sedikit perusahaan sektor industri memanfaatkan teknologi otomasi dalam bidang produksinya. Beberapa fungsi diterapkannya sistem otomasi industri yaitu dapat meningkatkan pendapatan, menghemat biaya, kuantitas dan kualitas yang lebih baik serta keamanan lebih terjaga. Salah satu aplikasi yang digunakan dalam proses otomasi yakni aplikasi sistem monitoring.

Penulis berinisiatif menggunakan sebuah mikrokontroler dan *LabVIEW* untuk memonitor *plant* yang telah dirancang,. Mikrokontroler atau biasa disebut dengan komputer mini merupakan piranti elektronik yang tersusun ke dalam sebuah chip *IC* dan berfungsi untuk melakukan suatu perintah atau tugas tertentu. Sedangkan *LabVIEW* adalah sebuah *software* milik National Instruments yang berfungsi untuk pemrosesan ataupun visualisasi data dalam bentuk grafis atau blok diagram pada bahasa pemrogramannya. Dengan memanfaatkan kedua komponen penting ini diharapkan dapat menampilkan data keluaran dari suatu *plant* secara *real time* tanpa harus kontak langsung untuk mengukur dan mencatat hasil *output* sebuah *plant* secara manual sehingga dapat mengefisiensikan waktu dan tenaga.[1] Pada penelitian kali ini penulis akan mengukur berapakah tegangan *AC*, tegangan *DC*, arus *AC*, arus *DC* dan daya yang dihasilkan dari 3 pembangkit pada laboratorium sistem ketenagaan yakni PLTMH, *solar cell* dan PLTB serta menampilkan hasil *outputnya* pada layar komputer (*LabVIEW*).

Dengan latar belakang tersebut maka penulis berinisiatif untuk mengambil sebuah judul Desain Sistem Monitoring 3 Pembangkit Hibrida pada Laboratorium Sistem Ketenagaan Universitas ISLAM Indonesia.

1.2 Rumusan Masalah

Adapun beberapa rumusan masalah yang didapatkan setelah menulis latar belakang sebagai berikut:

1. Bagaimana menentukan piranti apa saja yang diperlukan sehingga sistem monitoring dapat direalisasikan ke beberapa pembangkit listrik?
2. Bagaimana merancang sistem monitoring yang baik dan benar menggunakan *LabVIEW*?

1.3 Batasan Masalah

1. Menggunakan 3 pembangkit hibrida yang ada pada Laboratorium Sistem Ketenagaan Universitas ISLAM Indonesia yakni PLTMH, *solar cell* dan PLTB.
2. Memonitoring hasil keluaran 3 pembangkit yakni tegangan *AC*, tegangan *DC*, arus *AC*, arus *DC*, daya, frekuensi, faktor daya dan kecepatan putar pada turbin.
3. Menggunakan beban RLC load, baterai dan resistor keramik sebagai beban pengujian arusnya.

1.4 Tujuan Penelitian

Tujuan penelitian ini yaitu dapat merancang sistem monitoring kemudian merealisasikannya ke 3 buah pembangkit hibrida yang ada pada Laboratorium Sistem Ketenagaan Universitas Islam Indonesia menggunakan mikrokontroler arduino mega sebagai pembaca hasil *output* sensor dan laptop / komputer / *LabVIEW* sebagai *master*.

1.5 Manfaat Penelitian

1. Dapat menjadi referensi ilmu pengetahuan perancangan sistem monitoring.
2. Dapat memonitoring beberapa unit pembangkit listrik melalui ruang kontrol (laptop / komputer).

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Saat ini, sebagian besar Industri sedang bergeser menuju otomatisasi. Pada awalnya, pemantauan dan proses pengendalian dilakukan secara mandiri melalui pekerjaan manusia dan pastinya akan ada kemungkinan terjadinya *human error* saat mengukur suatu nilai parameter. Kurangnya pemantauan dan pengendalian dapat mempengaruhi lingkungan hidup pada manusia. Hal ini menjadi sebuah tantangan tugas untuk dapat melakukan pemantauan dan proses pengendalian beberapa parameter pada pembangkit listrik tenaga uap panas. Karena pembangkit harus bekerja 24 jam, jadi cukup sulit untuk memantau dan mengontrol beberapa parameter pada situs disetiap saat tanpa adanya masalah. Sistem pemantauan dan pengendalian sangat diperlukan dan memiliki peran yang cukup penting. Oleh karena itu kami mengusulkan sistem ini dengan menggunakan arduino dan representasi grafis dari *LabVIEW* untuk memantau dan mengontrol parameter seperti drum boiler level dan suhu uap. *LabVIEW* akan menampilkan representasi grafis, nilai akurat dan bentuk gelombang dari variasi parameter. Sistem ini terutama terdiri dari sensor suhu dan sensor ketinggian air. Semua data sensor diolah menggunakan arduino. Saat ada peningkatan atau penurunan tingkat secara tiba-tiba, modul *GSM (Global System for Mobile Communication)* digunakan untuk memberi informasi kepada para pejabat tinggi negara melalui *SMS (Short Message Service)*. [2]

Lain halnya dengan penelitian yang dilakukan oleh Abdurrahman, Heri Kusnadi dan Luki Utomo. Menurut beliau sangat baik jika di kehidupan mendatang dalam penyediaan tenaga listrik memiliki sifat yang ramah lingkungan dan dapat mengurangi polusi. Sinar matahari dapat dimanfaatkan sebagai sebuah pembangkit atau bisa disebut juga dengan *solar cell* (panel surya). Tentunya ketika melakukan proses pengukuran secara *real time* pasti mengalami kendala/kesulitan. Tujuan dari penelitian ini yaitu dapat mendesain dan mengaplikasikan sistem monitoring yang telah dibuat untuk mengukur berapa tegangan dan arus yang mengalir pada *solar cell*. Nilai tegangan dan arus pada *PV (Photovoltaic)* diperoleh melalui sensor yang terpasang, sensor tersebut berwujud dalam bentuk sinyal analog kemudian akan berubah ke dalam bentuk sinyal digital saat berada di mikrokontroler. Setelah melalui mikrokontroler sinyal digital tersebut akan diubah kembali menjadi ke dalam bentuk informasi grafis dengan tampilan *GUI (Graphics User Interface)* saat sudah berada pada *software LabVIEW*. Penelitian ini berlangsung selama 7 jam dengan menghasilkan daya *output* sebesar 40,6 - 142,6 watt dengan menggunakan 2x100 WP panel *PV*. Hasilnya dapat memonitor tegangan maupun arus yang mengalir secara *real time*. [3]

Konsep penelitian kali ini hampir sama dengan studi literatur yang telah ditemukan dimana tetap menggunakan arduino sebagai pembaca sensor dan *LabVIEW* sebagai penampil hasil datanya. Hanya saja perbedaan penelitian penulis terletak pada 3 poin penting. Pertama, sensor yang digunakan yakni menggunakan sensor *zmp101b* dan *voltage sensor*. Kedua, komunikasi antara arduino dengan *LabVIEW* menggunakan kabel *USB* sedangkan di salah satu studi literaturnya menggunakan komunikasi nirkabel yaitu dengan menggunakan *SMS*. Ketiga, objek yang dijadikan bahan penelitianpun berbeda yakni *PLTMH* dan *PLTB*.

2.2 Tinjauan Teori

Bagian ini terdiri atas teori-teori dasar yang mendukung dan saling terkait dengan penelitian yang sedang dilaksanakan.

2.2.1 Sistem Monitoring

Sistem monitoring adalah sebuah proses pengumpulan data dari berbagai sumber daya dan pada umumnya data tersebut merupakan data yang *real time*. Secara umum tujuan monitoring adalah untuk mendapatkan data-data atau pandangan agar diperoleh umpan balik bagi kebutuhan tertentu. Secara garis besar tahapan dalam sebuah sistem monitoring terbagi ke dalam tiga proses besar yaitu proses di dalam pengumpulan data monitoring, proses di dalam analisis data monitoring dan proses di dalam menampilkan data hasil monitoring.[4]

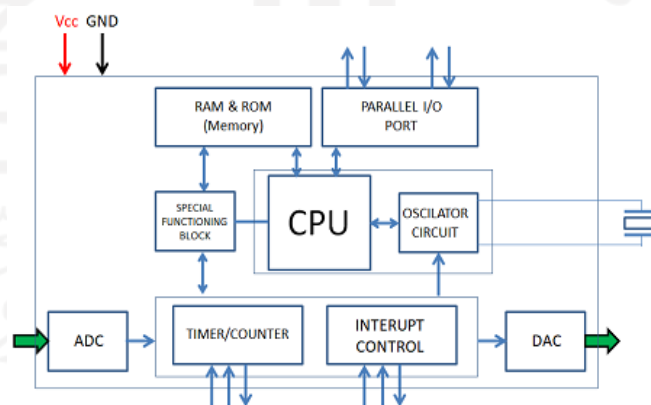
Kebanyakan kendala dan keterbatasan dalam melakukan monitoring terhadap objek yang hendak dipantau antara lain adalah: masih secara manual oleh petugas atau pemilik, masih menggunakan alat yang memberikan data pemantauan yang bersifat *general* dan harus dianalisis lebih lanjut, atau data pemantauan hanya dapat diamati lokal di terminal pemantauan saja. Hal ini menyebabkan banyak waktu dan upaya yang terbuang hanya untuk melakukan proses monitoring. Saat ini, teknologi yang dapat diterapkan untuk membangun suatu sistem monitoring sudah semakin maju dan berkembang. Teknologi ini dapat dimanfaatkan semaksimal mungkin untuk membantu manusia baik melakukan pemantauan terhadap suatu objek yang berada pada lokasi jarak jauh, maupun membantu manusia memonitor secara spesifik peristiwa atau kondisi yang hendak diantisipasi. Monitoring mengacu pada pengukuran kemajuan apakah masukan menghasilkan keluaran yang sudah sesuai dengan apa yang telah direncanakan.

2.2.2 Mikrokontroler

Mikrokontroler adalah suatu sistem komputer yang sebagian besar elemennya ataupun seluruhnya dikemas ke dalam sebuah chip *IC (Integrated Circuit)* menjadi satu bagian dari perangkat sistem yang lebih besar. Sehingga mikrokontroler sering disebut dengan *single chip microcomputer*. Berbeda halnya dengan *PC (Personal Computer)* yang mempunyai beberapa fungsi, mikrokontroler adalah sistem komputer yang memiliki satu atau beberapa tugas yang lebih detail/rinci.[5] Selain itu mikrokontroler mempunyai perbandingan *RAM* dan *ROM* yang berbeda dengan komputer. Mikrokontroler adalah sistem mikroprosesor yang dibuat langsung oleh pabriknya dan berisi peralatan internal seperti *RAM*, *ROM*, *CPU*, *I/O*, *clock* dan lainnya yang sudah saling terkait ataupun terhubung di dalam suatu chip *IC* yang sudah siap dipakai. Pada dasarnya mikrokontroler mempunyai *input* dan *output* serta kendali dengan program yang dapat ditulis dan dihapus secara khusus. Prinsip kerjanya hampir sama seperti membaca dan menulis data program. Sehingga kita hanya memprogram isi *ROM* sesuai aturan penggunaan oleh pabrik yang membuatnya. Mikrokontroler ditunjukkan pada Gambar 2.1 dan digram blok pada Gambar 2.2.



Gambar 2.1 Mikrokontroler



Gambar 2.2 Diagram Mikrokontroler

Berikut akan dijelaskan beberapa komponen penting yang ada di dalam sebuah mikrokontroler:[5]

1. CPU

CPU bisa dikatakan sebagai otak dari mikrokontroler karena dia bertanggungjawab untuk mengambil instruksi (*fetch*), menerjemahkannya (*decode*) dan kemudian dieksekusi (*execute*).

CPU merupakan suatu sistem yang terdiri dari bagian-bagian mikrokontroler yang saling terhubung. *CPU* memiliki fungsi utama yaitu mengambil instruksi lalu menerjemahkannya. *CPU* harus menerjemahkan intruksi tersebut yang diambil dari sebuah memori program.

2. Memori

Memori berfungsi untuk menyimpan data dan program yang telah dibuat. Di dalam mikrokontroler biasanya terdapat *RAM* dan *ROM* (*EPROM*, *EEPROM*, dan lain-lain) atau memori flash untuk menyimpan *source code program*.

3. Port *INPUT / OUTPUT* paralel

Port *I/O* berfungsi untuk menghubungkan berbagai perangkat agar saling terkoneksi seperti LED, *LCD*, memori, printer dan perangkat *I/O* lainnya ke mikrokontroller.

4. Port Serial (Serial Port)

Port serial menyediakan berbagai macam antarmuka serial antara mikrokontroler dengan periferal lain seperti port paralel.

5. Pengatur Waktu dan Penghitung (*Timer and Counter*)

Timer and Counter memiliki beberapa operasi utama didalam sebuah mikrokontroler antara lain pembangkitan pulsa, fungsi jam, osilasi, modulasi, pengukuran frekuensi, menghitung pulsa eksternal dan yang lainnya.

6. *Digital to Analog Converter* (*DAC*)

Konverter *DAC* berfungsi untuk mengubah sinyal digital ke bentuk sinyal analog. *DAC* digunakan untuk mengendalikan aplikasi analog seperti motor *DC* dan sebagainya.

7. *Analog to Digital Converter* (*ADC*)

ADC merupakan kebalikan dari *DAC* yaitu mengubah sinyal analog menjadi ke bentuk digital. Sinyal *input* dalam konverter ini harus dalam bentuk analog (misalnya *output* dari sensor) sedangkan *outputnya* dalam bentuk digital. *Output* digital biasanya digunakan untuk berbagai perangkat digital seperti layar digital pada perangkat pengukuran.

8. *Interrupt Control*

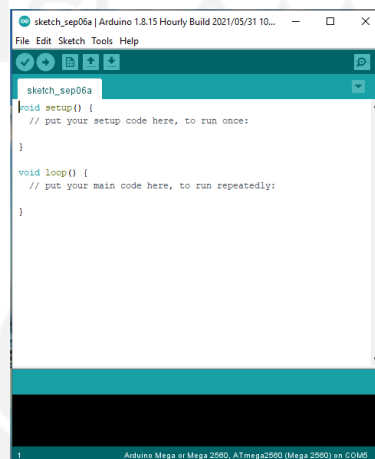
Interrupt Control berfungsi untuk menyediakan penundaan pada program kerja. *Interrupt* dapat berupa internal (menggunakan instruksi interupsi saat pemrograman) atau eksternal (diaktifkan menggunakan pin *interrupt*).

9. *Special Functioning Block*

Blok fungsi khusus merupakan pengontrol yang mempunyai beberapa port tambahan guna melaksanakan suatu fungsi tertentu. Pengontrol ini hanya dimiliki oleh beberapa mikrokontroler yang hanya bisa digunakan pada aplikasi tertentu seperti sistem robotik.

2.2.3 Arduino IDE

Arduino IDE merupakan sebuah *software* yang diciptakan menggunakan bahasa java dan berfungsi untuk mengcompiler dari bahasa C ke bahasa mesin mikrokontroler. Arduino IDE terlihat pada Gambar 2.3.



Gambar 2.3 Arduino IDE

Arduino IDE terdiri dari beberapa bagian, yaitu:[6]

1. Editor program yaitu berfungsi untuk menulis dan mengedit program.
2. Compiler yaitu sebuah modul yang berfungsi untuk mengubah kode program (bahasa processing) menjadi kode biner, karena mikrokontroler hanya bisa memahami bahasa dengan kode biner.
3. Uploader sebagai sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan arduino. Arduino menggunakan pemrograman dengan bahasa C.

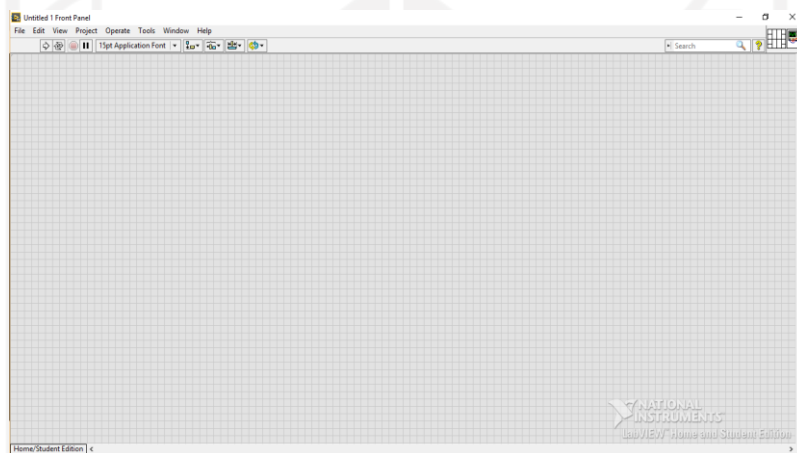
2.2.4 LabVIEW

LabVIEW atau *Laboratory Virtual Instrumentation Engineering Workbench* merupakan *software* pemrograman berbasis grafis atau blok diagram yang dimiliki oleh *National Instruments*. *LabVIEW* memiliki fungsi dan peranan yang tidak jauh berbeda dengan *software* pemrograman lainnya seperti *matlab* atau *visual basic*, C++ dan sejenisnya. Perbedaan *software LabVIEW* dengan C++ ataupun *matlab* terdapat pada bahasa pemrogramannya yaitu menggunakan grafis atau blok diagram untuk *LabVIEW* sedangkan basis teks untuk *software* pemrograman lainnya. Karena bahasa pemrograman pada *LabVIEW* hampir sama dengan sebuah *instrumen* baik dari sisi

tampilan maupun pengoperasiannya maka program *LabVIEW* disebut dengan istilah *Virtual Instrumens* atau *VI*. Hal pertama yang dilakukan saat akan membuat program pada *LabVIEW* yaitu membuat front panel atau *user interface* menggunakan *control* dan *indicator*. *Control* disini dapat memuat *dials*, *knobs*, *push buttons* dan peralatan *input* lainnya sedangkan *indicator* memuat *leds*, *graphs* dan peralatan *display* lainnya. Kemudian langkah kedua yaitu merancang blok diagram yang memuat kode-kode *VI* guna mengatur *user interface*. *LabVIEW* sendiri tersusun atas 3 komponen penting yaitu:[7]

1. Front Panel

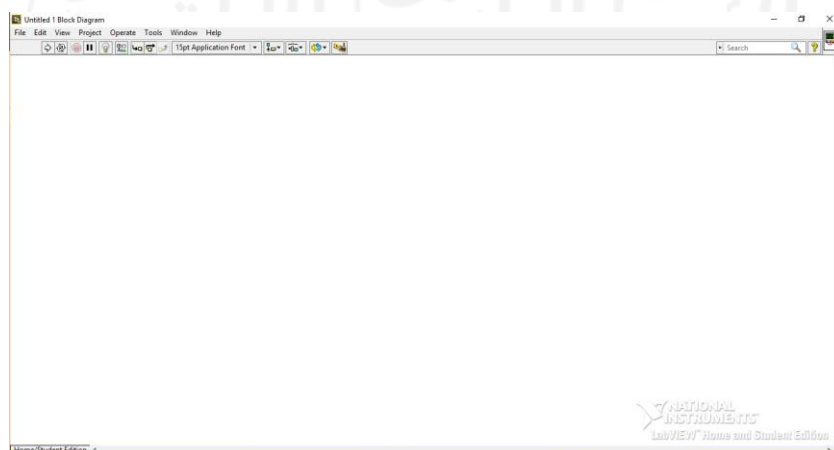
Front panel merupakan bagian window yang berisi *indicator* dan *control* dengan latar belakang berwarna abu-abu. Fungsi front panel untuk merancang/mendirikan *VI*, menjalankan program dan mendebug program. Front panel ditunjukkan pada Gambar 2.4.



Gambar 2.4 Front Panel

2. Blok Diagram dari VI

Blok diagram merupakan bagian dari window dengan latar belakang berwarna putih yang berisi *source code* untuk arahan pada front panel. Block diagram ditunjukkan pada Gambar 2.5.



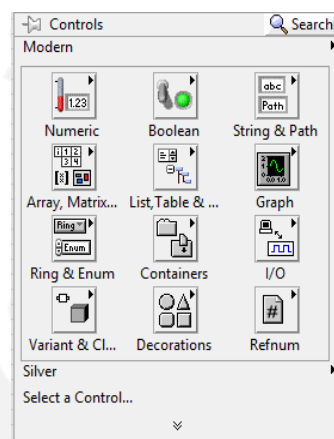
Gambar 2.5 Block Diagram

3. Control Pallette dan Functions Pallette

Control dan functions pallette memiliki fungsi sebagai penyusun dari sebuah VI.

a) Control Pallette

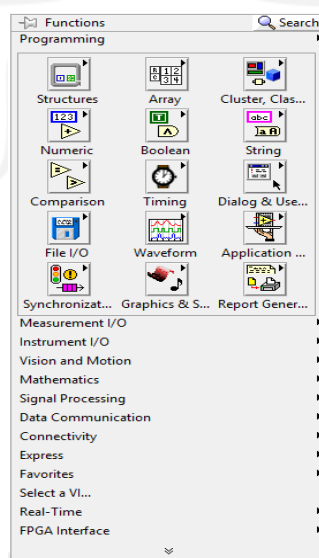
Control pallette hanya tersedia pada front panel yang berisi tentang beberapa indikator maupun kontrol. Adapun cara untuk menampilkan control pallette dengan klik kanan pada front panel tersebut atau dengan mengklik windows => *show control pallette*. Control pallette dapat dilihat di Gambar 2.6.



Gambar 2.6 Control Pallette

b) Function Palette

Functions pallette hanya tersedia pada blok diagram dan berfungsi untuk merancang sebuah blok diagram. Adapun cara untuk menampilkan functions pallette dengan klik kanan pada lembar kerja blok diagram tersebut atau dengan mengklik windows => *show control*. Penampilan functions pallette dapat dilihat di Gambar 2.7.



Gambar 2.7 Functions Pallette

BAB 3

METODOLOGI

Bagian ini menjelaskan perancangan sistem yang digunakan, cara mengimplementasikan rancangan dan cara pengujian sistem (indikator kinerja dan cara mengukurnya). Penjelasan ini terdiri dari beberapa bab yang saling terkait.

3.1 Alat dan Bahan

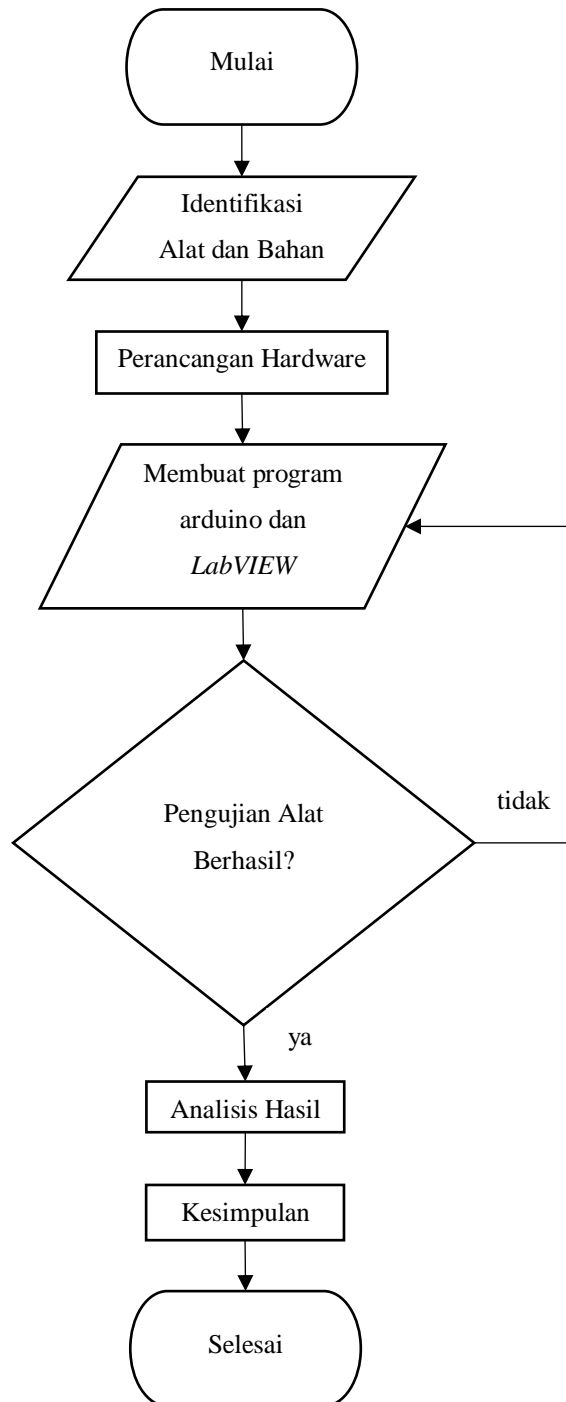
Berikut alat dan bahan yang diperlukan selama penelitian berlangsung, antara lain:

1. Arduino mega
2. *Software* arduino
3. *Software* LabVIEW
4. *Solar cell*
5. PLTB
6. PLTMH
7. Blower
8. *SCC (Solar Cell Controller)*
9. Voltage sensor
10. Sensor zmpt101b
11. Sensor acs712
12. Tachometer
13. Baterai 7,2 Ah
14. Beban RLC
15. Cosphi meter
16. Multimeter
17. Kabel *swith hub*
18. Kabel jumper *male to female*
19. Kabel serabut dan tunggal

3.2 Alur Penelitian

Beberapa proses yang diperlukan agar alat monitoring ini dapat dirancang yaitu dimulai dengan mengidentifikasi alat dan bahan yang diperlukan apa saja, perancangan hardware, membuat program pada arduino ide dan *LabVIEW*, melakukan pengujian alat apakah berhasil atau

tidak. Jika berhasil membaca output dari 3 pembangkit hibrida maka akan dilakukan pencatatan hasil pengujian dan analisa hasil. Kemudian mengambil sebuah kesimpulan terhadap alat monitoring yang dirancang dan data yang sudah dianalisa. Alur penelitian dapat dilihat pada flowchart penelitian Gambar 3.1.



Gambar 3.1 Flowchart Penelitian

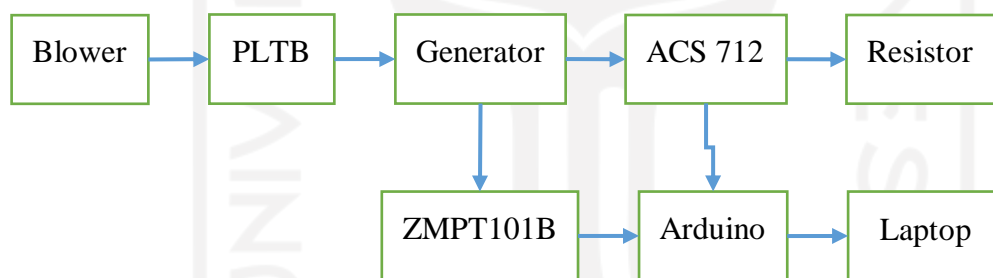
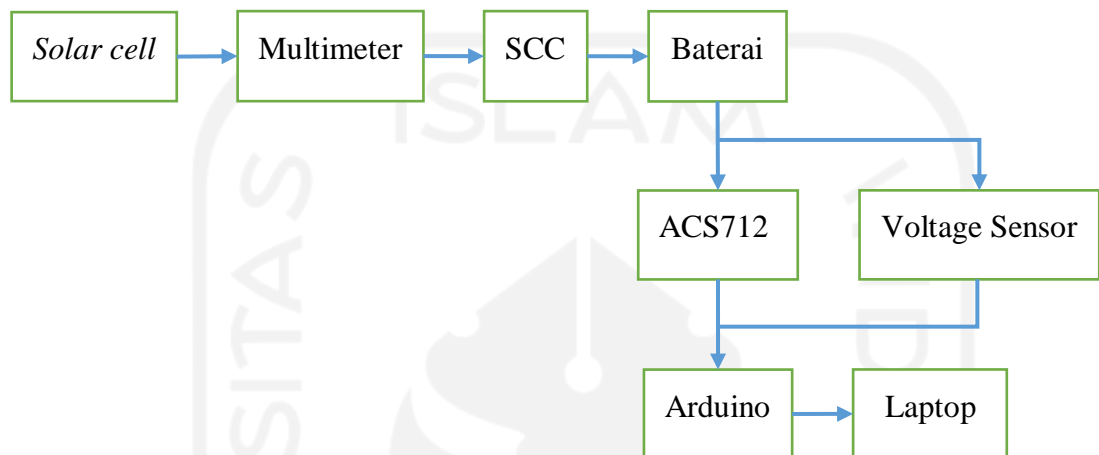
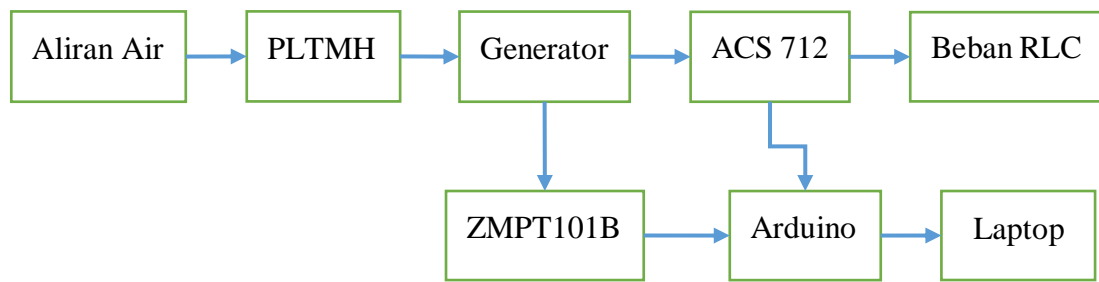
Urgensi dari sistem monitoring ini adalah untuk mengetahui performa yang dihasilkan dari 3 buah pembangkit hibrida pada Laboratorium Sistem Ketenagaan Universitas ISLAM Indonesia. 3 pembangkit hibrida tersebut yakni PLTMH (Pembangkit Listrik Tenaga Mikro Hidro), *Solar*

Cell dan PLTB (Pembangkit Listrik Tenaga Bayu). Adapun target yang ingin dicapai dari sistem monitoring ini adalah dapat memantau performa dari setiap pembangkit secara *real time* dari waktu ke waktu sehingga data tersebut nantinya dapat dianalisa. Pemantauan dapat dilakukan secara otomatis dari jarak jauh tanpa harus kontak langsung di lokasi pembangkit sehingga dapat mengefisiensikan waktu dan tenaga.

Beberapa parameter yang dipilih dalam sistem monitoring ini antara lain mengetahui berapa tegangan AC dan arus AC baik dari PLTMH maupun PLTB serta mengetahui berapa tegangan DC, arus DC dan daya DC pada *Solar Cell*. Proses monitoring dilakukan dengan memanfaatkan arduino mega sebagai mikrokontroler pembaca hasil output dari sensor yang digunakan dan LabVIEW sebagai penampil data yang disajikan. Adapun sensor yang digunakan yaitu zmpt101b untuk mengukur tegangan AC, sensor voltage untuk mengukur tegangan DC serta acs712 untuk mengukur arus AC dan DC pada 3 pembangkit hibrida pada Laboratorium Sistem Ketenagaan Universitas ISLAM Indonesia. Beberapa parameter tersebut digunakan sebagai data yang akan diproses hingga menjadi suatu informasi yang dapat digunakan oleh user atau masyarakat umum.

3.3 Proses Pengiriman Data Sistem Monitoring

Pengiriman data pada penelitian kali ini menggunakan komunikasi serial. Data yang dihasilkan dari 3 pembangkit hibrida yakni PLTMH, *solar cell* maupun PLTB merupakan data analog. Kemudian dari data analog ini berjalan menuju ke sensor zmpt101b, voltage sensor, acs712 dan ke arduino mega yang telah diprogram menggunakan software arduino IDE. Dari arduino mega data diolah menjadi data digital atau disebut dengan proses *ADC (Analog to Digital Converter)* karena pada dasarnya arduino hanya dapat membaca bahasa mesin dengan kode biner. Data lalu dibaca oleh analogread kemudian diterjemahkan ke nilai tegangan AC, tegangan DC, arus AC dan arus DC yang sesungguhnya. Setelah itu nilai tegangan AC, tegangan DC, arus AC dan arus DC berlanjut menuju ke *LabVIEW* untuk ditampilkan pada front panel baik dalam bentuk grafik maupun angka. Perancangan sistem monitoring dapat dilihat pada blok diagram Gambar 3.2



Gambar 3.2 Diagram Blog Penelitian

3.3.1 Sensor

Pada penelitian kali ini kita menggunakan 3 sensor yaitu voltage sensor, zmpt101b dan sensor arus acs712, berikut penjelasannya:

1. Voltage Sensor

Voltage sensor merupakan piranti yang digunakan untuk mendeteksi nilai suatu tegangan. Prinsip yang digunakan berdasarkan penekanan resistansi yang dapat menurunkan tegangan *inputnya* 5 kali lebih kecil dari tegangan aslinya. Voltage sensor dapat dilihat pada Gambar 3.3.



Gambar 3.3 Voltage Sensor

Nilai maksimal range yang dapat diolah adalah 5 VDC. Arduino hanya menerima data berupa nilai *ADC*, agar dapat menentukan tegangan maka menggunakan persamaan (3.1):

$$\text{Tegangan} = \frac{ADC}{1023 \times 5V} \quad (3.1)$$

Keterangan:

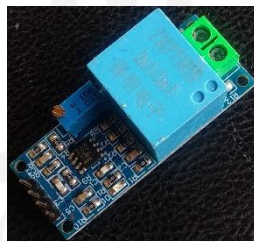
ADC: nilai *ADC* yang terbaca

1023: nilai maksimal *ADC* yaitu 10 bit

5V: Tegangan referensi *ADC* arduino

2. ZMPT101B

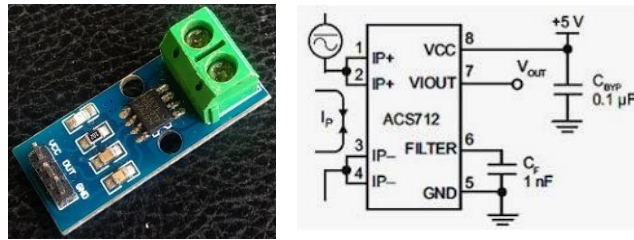
ZMPT101B merupakan sebuah piranti elektronik yang berfungsi untuk mengukur tegangan AC 1 fasa antara 110-250 VA dengan menggunakan prinsip transformator step-down dan sensor ini dapat secara langsung dihubungkan ke tegangan AC PLN.[8] Tampilan zmpt101b dapat dilihat pada Gambar 3.4.



Gambar 3.4 Sensor ZMPT101B

3. ACS712

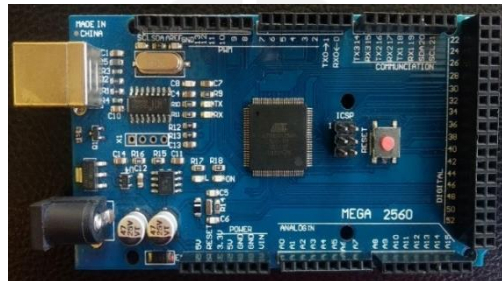
ACS712 merupakan piranti elektronik yang berfungsi untuk mengetahui arus AC maupun DC yang mengalir pada sebuah kabel. Sensor ini bekerja berdasarkan *hall effect* yaitu dengan memanfaatkan medan magnetik disekitar arus, yang dibaca pada kabel tembaga kemudian dikonversi menjadi sinyal proporsional.[8] Sinyal tersebut dapat berbentuk tegangan data digital maupun tegangan analog. Kemudian sinyal tersebut akan difungsikan menjadi alat ukur arus yang dapat disimpan ke dalam sebuah penyimpanan seperti server untuk dianalisa maupun sebagai alat kontrol. Tampilan sensor acs712 dan skematiknya ditunjukkan pada Gambar 3.5.



Gambar 3.5 Sensor ACS712 dan Skematik ACS712

3.3.2 Arduino Mega

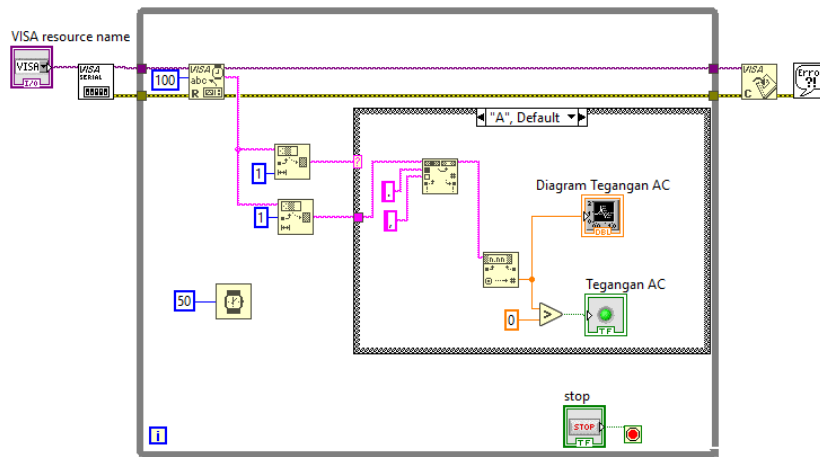
Arduino mega merupakan papan mikrokontroler yang sudah dirancang dengan beberapa komponen pelengkap seperti tombol reset, digital *input* dan perangkat lainnya serta dapat dikoneksikan dengan komputer menggunakan kabel *USB*. Tampilan arduino mega dapat dilihat pada Gambar 3.6.



Gambar 3.6 Arduino Mega

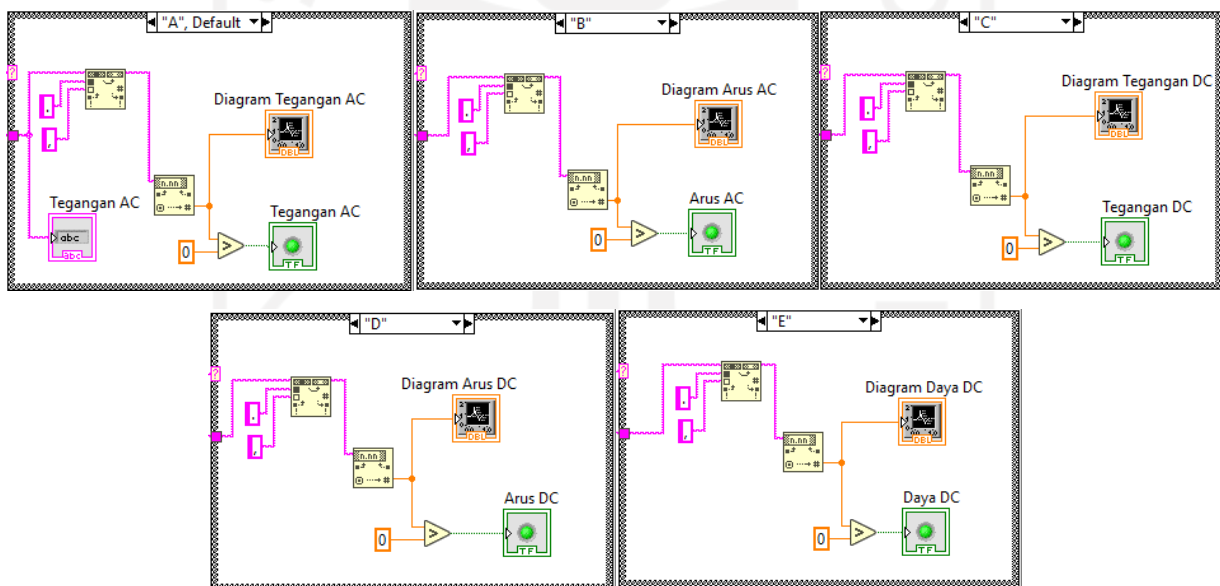
3.3.3 Laptop

Laptop berfungsi untuk memonitor hasil dari pembacaan arduino terhadap sensor yang terpasang menggunakan *software LabVIEW*. Komunikasi antara laptop dan arduino menggunakan sebuah alat yaitu *switch hub*. *Switch hub* berfungsi sebagai penghubung kabel-kabel *LAN* untuk menerima atau mengirimkan data dari laptop ke arduino maupun sebaliknya. Tampilan block diagram *LabVIEW* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Tampilan Monitoring Pada Block Diagram *LabVIEW*

Pada block diagram ini digunakan case structure. Case structure berfungsi untuk mengeksekusi satu kasus sesuai kondisi inputnya. Nilai input akan memutuskan case mana yang akan mengeksekusi. Di bagian atas case structure terdapat label selector yang berisi case atau kasus. Klik panah ke bawah untuk melihat keseluruhan kasus yang tersedia. Klik panah ke kanan untuk melihat kasus selanjutnya dan klik panah ke kiri untuk melihat kasus sebelumnya. Case structure ditunjukkan pada Gambar 3.8.

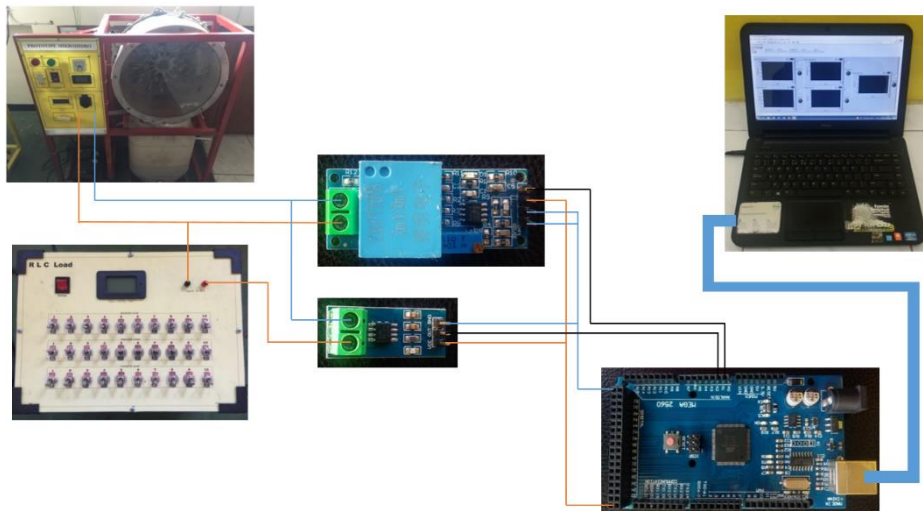


Gambar 3.8 Case Structure

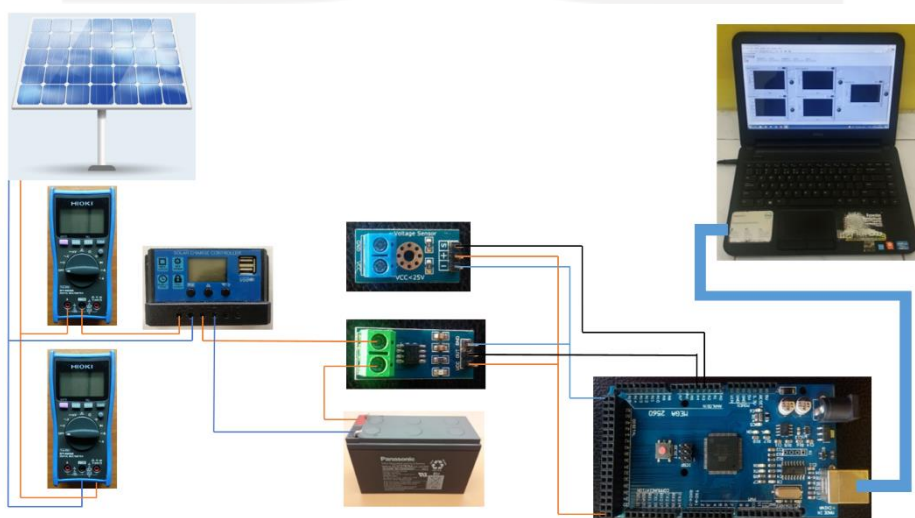
Ada 5 kasus yang telah diprogram pada blok diagram ini. Kasus dengan simbol A menunjukkan untuk mengeksekusi kondisi input dari tegangan AC. Kasus dengan simbol B menunjukkan untuk mengeksekusi kondisi input dari arus AC. Kasus dengan simbol C menunjukkan untuk mengeksekusi kondisi input dari tegangan DC. Kasus dengan simbol D menunjukkan untuk mengeksekusi kondisi input dari arus DC. Kasus dengan simbol E menunjukkan untuk mengeksekusi kondisi input dari daya DC. Tipe data yang mengalir ke dalam case structure ini berupa data string.

3.4 Rangkaian Pengujian Sistem Monitoring

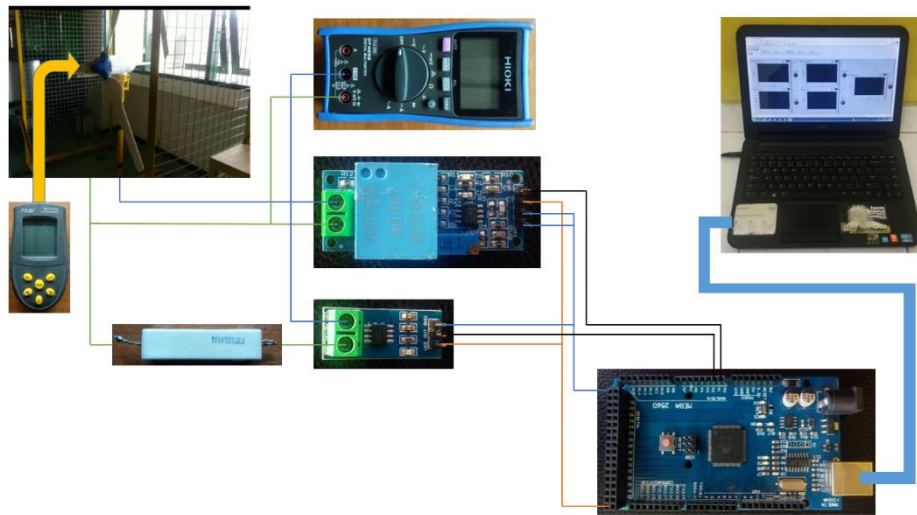
Rangkaian pengujian sistem monitoring diperlukan untuk melihat kabel-kabel yang terhubung dari 3 pembangkit hibrida hingga ke *LabVIEW*. Kabel yang digunakan yakni kabel jumper *male to female*, kabel *switch hub* dan kabel tunggal / serabut. Kabel tunggal / serabut digunakan untuk *output* dari 3 pembangkit hibrida. Kabel *male to female* digunakan untuk komunikasi dari sensor ke arduino mega. Kabel *male to female* pada sensor terhubung ke beberapa pin di arduino mega yakni sensor *zmpt101b* terhubung ke pin A0, *acs712* arus AC terhubung ke pin A1, *voltage sensor* terhubung ke pin A2 dan *acs712* arus DC terhubung ke pin A3. Kemudian komunikasi antara arduino mega dengan *LabVIEW* terhubung dengan kabel *switch hub*. Rangkaian pengujian sistem monitoring pada 3 pembangkit hibrida dapat dilihat pada Gambar 3.9, Gambar 3.10 dan Gambar 3.11.



Gambar 3.9 Rangkaian Pengujian PLTMH



Gambar 3.10 Rangkaian Pengujian Solar Cell



Gambar 3.11 Rangkaian Pengujian PLTB

3.5 Cara Analisis

Analisis pertama dilakukan untuk menguji karakteristik setiap sensor yang digunakan yakni *zmpt101b*, *acs712* dan *voltage sensor* yang dihubungkan dengan *arduino mega* dan telah diprogram melalui *software arduino IDE*. Rumus yang digunakan untuk mencari persentase *error* dan persentase keberhasilan ditulis dengan persamaan (3.2) dan (3.3):

$$\%Error = \left| \frac{\text{Nilai Aktual} - \text{Nilai Terukur}}{\text{Nilai Aktual}} \right| \times 100\% \quad (3.2)$$

$$\%Keberhasilan = 100\% - \%Error \quad (3.3)$$

Analisis kedua dilakukan dengan melihat pengaruh beban RLC pada output PLTMH baik dari tegangan AC, arus AC dan frekuensi serta *error* yang didapatkan dari pengiriman data serial monitor ke *LabVIEW*. Analisis ketiga dilakukan guna menguji berapa lama baterai akan terisi penuh ketika mendapat pasokan tegangan DC maupun arus DC dari *solar cell* serta *error* yang didapatkan dari pengiriman data serial monitor ke *LabVIEW*. Rumus yang digunakan untuk mencari berapa lama baterai akan terisi penuh ditulis dengan persamaan (3.4)[9].

$$T_a = \frac{C}{I} (1 + 20\%) \quad (3.4)$$

Keterangan:

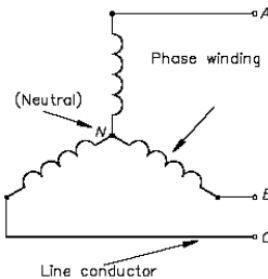
T_a = Lamanya pengisian arus pada baterai (h / hour)

C = Kapasitas baterai (Ah / Ampere hour)

I = Arus pengisian baterai (A / Ampere)

Analisis keempat yakni mencari daya yang dihasilkan dari generator AC 3 fasa pada PLTB serta *error* yang didapatkan dari pengiriman data serial monitor ke *LabVIEW*. Generator yang digunakan menggunakan rangkaian hubungan bintang (tertera pada Gambar 3.12). Pada hubungan

bintang (Y, wye), ujung-ujung tiap fase dihubungkan menjadi satu dan menjadi titik netral atau titik bintang. Tegangan antara dua terminal (V_{Line}) dari tiga terminal A – B – C mempunyai besar magnitude dan beda fasa yang berbeda dengan tegangan tiap terminal terhadap titik netral (V_{fasa}). Tegangan V_a , V_b dan V_c disebut tegangan “fasa” atau V_f .



Gambar 3.12 Hubungan Bintang

Tegangan yang mengalir pada tiap fasanya seimbang namun memiliki perbedaan fasa yang berbeda sebesar 120 derajat. Pada rangkaian bintang ini berlaku persamaan (3.5) dan (3.6):

$$V_{fasa} = \sqrt{3} \cdot V_{Line} \quad (3.5)$$

$$I_R = I_S = I_T = I_{Line} \quad (3.6)$$

Guna menghitung kapasitas daya generatornya menggunakan persamaan (3.7):

$$P = (V_{Line R} \cdot I_R \cdot \cos\phi) + (V_{Line S} \cdot I_S \cdot \cos\phi) + (V_{Line T} \cdot I_T \cdot \cos\phi) \quad (3.7)$$

dengan nilai:

P = Daya generator 3 fasa (watt)

$V_{Line R,S,T}$ = Tegangan fasa R-N, fasa S-N, fasa T-N (volt)

$I_{R,S,T}$ = Arus fasa R, fasa S, fasa T (amper)

$\cos\phi$ = Faktor daya

Kemudian analisis terakhir adalah mengetahui penilaian *user* terhadap tampilan dan fungsi sistem monitoring yang telah dibuat dengan cara melakukan survey terhadap 5 orang yang berbeda menggunakan kuisisioner dari *google form*.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Karakteristik Sensor ZMPT101B

Pengujian karakteristik sensor *zmpt101b* dilakukan dengan membandingkan keluaran *voltage regulator* dengan *zmpt101b*. Karakteristik sensor *zmpt101b* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Karakteristik Sensor ZMPT101B

Voltage Regulator (V)	ZMPT101B (V)	Error (%)	Keberhasilan (%)
0	0	0	100
10	9,29	7,1	92,9
20	19,3	3,5	96,5
30	29,31	2,3	97,7
40	39,34	1,65	98,35
50	49,46	1,08	98,92
60	59,54	0,766667	99,233333
70	69,56	0,628571	99,371429
80	79,57	0,5375	99,4625
90	89,63	0,411111	99,588889
100	99,65	0,35	99,65
\bar{X} Error (%) dan \bar{X} Keberhasilan (%)		1,665804	98,334196

Hasil perbandingan dari pengujian sensor yang telah dibuat dengan *voltage regulator* didapatkan persentase rata-rata keberhasilannya adalah 98,33 %.

4.2 Karakteristik Voltage Sensor

Pengujian karakteristik *voltage sensor* dilakukan dengan membandingkan keluaran *DC power supply* dengan *voltage sensor*. Karakteristik *voltage sensor* dapat dilihat pada Tabel 4.2

Tabel 4.2 Karakteristik Voltage Sensor

DC Power Supply (V)	Voltage Sensor (V)	Error (%)	Keberhasilan (%)
13,34	13,32	0,149925	99,850075
13,37	13,37	0	100
13,66	13,65	0,073206	99,926794
14,25	14,25	0	100
14,26	14,25	0,070126	99,929874
14,32	14,32	0	100
14,36	14,35	0,069638	99,930362
14,43	14,42	0,0693	99,9307
14,64	14,62	0,136612	99,863388
15,11	15,1	0,066181	99,933819
15,5	15,5	0	100
\bar{X} Error (%) dan \bar{X} Keberhasilan (%)		0,057726	99,942274

Hasil perbandingan dari pengujian sensor yang telah dibuat dengan *DC power supply* didapatkan persentase rata-rata keberhasilannya adalah 99,94 %.

4.3 Karakteristik Sensor ACS712 Arus AC

Pengujian karakteristik sensor acs712 dilakukan dengan membandingkan keluaran PLTMH dengan acs712. Karakteristik sensor acs712 arus AC dapat dilihat pada Tabel 4.3.

Tabel 4.3 Karakteristik Sensor ACS712 Arus AC

AC Current PLTMH (A)	AC Current ACS712 (A)	Error (%)	Keberhasilan (%)
0,07	0,05	28,571429	71,42857143
0,1	0,09	10	90
0,16	0,13	18,75	81,25
0,19	0,17	10,526316	89,47368421
0,24	0,21	12,5	87,5
0,34	0,33	2,9411765	97,05882353
0,38	0,37	2,6315789	97,36842105
0,46	0,45	2,173913	97,82608696
0,48	0,45	6,25	93,75
0,52	0,49	5,7692308	94,23076923
0,65	0,65	0	100
0,84	0,81	3,5714286	96,42857143
0,88	0,85	3,4090909	96,59090909
0,95	0,93	2,1052632	97,89473684
0,99	0,97	2,020202	97,97979798
\bar{X} Error (%) dan \bar{X} Keberhasilan (%)		7,414641883	92,58535812

Hasil perbandingan dari pengujian sensor yang telah dibuat dengan output PLTMH didapatkan persentase rata-rata keberhasilannya adalah 92,19 %.

4.4 Karakteristik Sensor ACS712 Arus DC

Pengujian karakteristik sensor acs712 dilakukan dengan membandingkan keluaran DC power supply dengan acs712. Karakteristik sensor acs712 arus DC dapat dilihat pada Tabel 4.4.

Tabel 4.4 Karakteristik Sensor ACS712 Arus DC

DC Current Regulator (A)	DC Current ACS712 (A)	Error (%)	Keberhasilan (%)
0,19	0,15	21,05263	78,94737
0,24	0,2	16,66667	83,33333
0,36	0,35	2,777778	97,222222
0,38	0,35	7,894737	92,105263
0,43	0,4	6,976744	93,023256
0,45	0,45	0	100
0,46	0,45	2,173913	97,826087
0,52	0,5	3,846154	96,153846
0,54	0,5	7,407407	92,592593
0,56	0,55	1,785714	98,214286
\bar{X} Error (%) dan \bar{X} Keberhasilan (%)		7,058175	92,941825

Hasil perbandingan dari pengujian sensor yang telah dibuat dengan DC power supply didapatkan persentase rata-rata keberhasilannya adalah 92,94 %.

4.5 Pengujian Beban RLC Terhadap V, I, f dan $\cos\phi$ Pada PLTMH

Pengambilan data dilakukan pada tanggal 2 Maret 2021, dengan cara menyambungkan output PLTMH ke beban RLC dan alat ukur.

4.5.1 Data Pengujian PLTMH Dengan Beban Resistif

Pengujian dilakukan dengan menyambungkan kabel keluaran PLTMH ke beban R. Berikut hasil perolehan datanya dapat dilihat pada Tabel 4.5.

Tabel 4.5 Pengujian PLTMH Dengan Beban Resistif

Beban R (Ω)	Tegangan AC (V)		Error (%)	Arus AC (A)		Error (%)	Frekuensi (Hz)
	Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW		
97	41	40,9	0,24	0,39	0,38	2,56	60,5
102	41	40,88	0,29	0,37	0,36	2,7	60,4
114	41	40,85	0,37	0,34	0,33	2,94	60,4
125	41	40,91	0,22	0,32	0,31	3,13	60,3
137	41	41	0	0,29	0,29	0	60,3
151	41	40,86	0,34	0,27	0,27	0	60,5
164	41	40,84	0,39	0,25	0,24	4	60,3
182	41	41	0	0,23	0,23	0	60,5
190	41	40,87	0,32	0,22	0,22	0	60,5
204	41	40,89	0,27	0,21	0,21	0	60,4
	\bar{X} Error (%)		0,24	\bar{X} Error (%)		1,53	

Dari data yang diperoleh terjadi perbedaan hasil, antara data yang tertampil di serial monitor dengan *LabVIEW*. Persentase rata-rata *error* yang dihasilkan pada pengujian PLTMH dengan beban resistif yakni memperoleh nilai 0,24% untuk tegangan AC dan 1,53% untuk arus AC. Nilai rata-rata persentase kesalahan yang diperoleh tergolong baik karena nilainya masih dibawah 5% sesuai dengan SNI. Kemungkinan *error* terjadi karena nilai output dari PLTMH yang terkadang berubah-ubah (tidak konstan) ataupun bisa dari sensor-sensor yang digunakan.

Frekuensi diatur mendekati pada angka 60 Hz ketika nilai beban berubah, sehingga menghasilkan data tersebut. Sesuai kaidah hukum ohm yaitu semakin kecil nilai resistif yang digunakan, maka semakin besar nilai arus yang mengalir di suatu penghantar pada rangkaian tertutup dengan tegangan yang tetap.

4.5.2 Data Pengujian PLTMH Dengan Beban Induktif

Pengujian dilakukan dengan menyambungkan kabel keluaran mikrohidro ke beban L. Berikut hasil perolehan datanya yang dapat dilihat pada

Tabel 4.6.



Tabel 4.6 Pengujian PLTMH Dengan Beban Induktif

Beban L (mH)	Tegangan AC (V)		Error (%)	Arus AC (A)		Error (%)	Frekuensi (Hz)
	Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW		
230	84	83,87	0,16	0,58	0,56	3,45	100,2
271	80	79,85	0,19	0,47	0,46	2,13	100
316	78	78	0	0,39	0,38	2,56	100,6
392	74	74	0	0,30	0,3	0	100,1
513	71	70,96	0,056	0,22	0,22	0	100,1
674	68	67,85	0,22	0,16	0,15	6,25	100,3
871	66	65,89	0,17	0,12	0,11	8,33	100,5
910	63	62,9	0,16	0,11	0,11	0	100,2
	\bar{X} Error (%)		0,12	\bar{X} Error (%)		2,84	

Dari data yang diperoleh terjadi perbedaan hasil, antara data yang tertampil di serial monitor dengan *LabVIEW*. Persentase rata-rata *error* yang dihasilkan pada pengujian PLTMH dengan beban induktif yakni memperoleh nilai 0,12% untuk tegangan AC dan 2,84% untuk arus AC. Nilai rata-rata persentase kesalahan yang diperoleh tergolong baik karena nilainya masih dibawah 5% sesuai dengan SNI. Kemungkinan *error* terjadi karena nilai output dari PLTMH yang terkadang berubah-ubah (tidak konstan) ataupun bisa dari sensor-sensor yang digunakan.

Frekuensi diatur mendekati pada angka 100 Hz ketika nilai beban berubah, sehingga menghasilkan data tersebut. Semakin besar nilai beban induktifnya, maka semakin kecil nilai tegangan dan arus yang mengalir.

4.5.3 Data Pengujian PLTMH Dengan Beban Kapasitif

Pengujian dilakukan dengan menyambungkan kabel keluaran PLTMH ke beban C. Berikut hasil perolehan datanya yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Pengujian PLTMH Dengan Beban Kapasitif

Beban C (mF)	Tegangan AC (V)		Error (%)	Arus AC (A)		Error (%)	Frekuensi (Hz)
	Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW		
1554	54	53,88	0,22	0,37	0,36	2,70	70,2
1741	56	55,87	0,23	0,43	0,43	0	70,2
2029	57	56,91	0,16	0,51	0,50	1,96	70,2
2306	60	60	0	0,61	0,60	1,64	70,2
2634	62	61,9	0,16	0,72	0,72	0	70,2
2966	65	64,95	0,077	0,85	0,84	1,18	70,2
	\bar{X} Error (%)		1,14	\bar{X} Error (%)		1,25	

Dari data yang diperoleh terjadi perbedaan hasil, antara data yang tertampil di serial monitor dengan *LabVIEW*. Persentase rata-rata *error* yang dihasilkan pada pengujian PLTMH dengan beban kapasitif yakni memperoleh nilai 1,14% untuk tegangan AC dan 1,25% untuk arus AC. Nilai rata-rata persentase kesalahan yang diperoleh tergolong baik karena nilainya masih dibawah 5%

sesuai dengan SNI. Kemungkinan *error* terjadi karena nilai output dari PLTMH yang terkadang berubah-ubah (tidak konstan) ataupun bisa dari sensor-sensor yang digunakan.

Frekuensi diatur mendekati pada angka 70 Hz ketika nilai beban berubah, sehingga menghasilkan data tersebut. Semakin besar nilai beban kapasitifnya, maka semakin besar nilai tegangan dan arus yang mengalir.

4.6 Pengujian Solar Cell

Pengujian dilakukan dengan menyambungkan output *solar cell* ke SCC kemudian dari SCC kabel menuju ke baterai. SCC (*Solar Charge Controller*) berfungsi untuk mengatur kelebihan tegangan DC atau arus DC berlebih yang mengalir dari *solar cell* ke baterai agar dapat mengoptimalkan sistem dan menjaga masa pakai baterai bertahan lebih lama. [9] Kapasitas baterai yang digunakan bernilai 7,2 Ah. Berikut hasil perolehan datanya yang dapat dilihat pada Tabel 4.8 dan Tabel 4.9.

Tabel 4.8 Pengukuran Pada Output Solar Cell

No	Jam	Tegangan DC (V)			Arus DC (A)			Daya (Watt)		
		Serial Monitor Arduino	LabVIEW	Error (%)	Serial Monitor Arduino	LabVIEW	Error (%)	Serial Monitor Arduino	LabVIEW	Error (%)
1	10.20	14,43	14,41	0,14	0,43	0,42	2,33	6,21	6,05	2,58
2	10.25	13,35	13,30	0,37	0,36	0,35	2,78	4,80	4,66	2,92
3	10.30	13,41	13,39	0,15	0,37	0,37	0	4,96	4,95	0,20
4	10.35	14,2	14,19	0,07	0,4	0,39	2,5	5,68	5,53	2,64
5	10.40	15,1	15,08	0,13	0,51	0,5	1,96	7,70	7,54	2,08
6	10.45	15,2	15,18	0,13	0,52	0,5	3,85	7,90	7,59	3,93
7	10.50	14,63	14,58	0,34	0,46	0,46	0	6,73	6,71	0,30
8	10.55	14,5	14,47	0,21	0,44	0,42	4,55	6,38	6,08	4,70
9	11.00	15,5	15,47	0,19	0,56	0,54	3,57	8,68	8,35	3,80
10	11.05	15,34	15,3	0,26	0,53	0,53	0	8,13	8,11	0,25
11	11.10	15,45	15,41	0,26	0,54	0,53	1,85	8,34	8,17	2,04
12	11.15	15,27	15,25	0,13	0,52	0,52	0	7,94	7,93	0,13
13	11.20	15,16	15,13	0,2	0,51	0,51	0	7,73	7,72	0,13
14	11.25	14,55	14,5	0,34	0,47	0,46	2,13	6,84	6,67	2,49
15	11.30	14,36	14,3	0,42	0,43	0,43	0	6,17	6,15	0,32
\bar{X} Solar Cell		14,69	14,66	0,22	0,47	0,46	1,70	6,90	6,74	1,90

Tabel 4.9 Pengukuran Pada Baterai

No	Jam	Tegangan DC (V)			Arus DC (A)			Daya (Watt)		
		Serial Monitor Arduino	LabVIEW	Error (%)	Serial Monitor Arduino	LabVIEW	Error (%)	Serial Monitor Arduino	LabVIEW	Error (%)
1	10.20	14,2	14,17	0,21	0,31	0,3	3,23	4,40	4,25	3,41
2	10.25	13,3	13,26	0,30	0,19	0,18	5,26	2,53	2,39	5,53
3	10.30	13,37	13,37	0,00	0,19	0,18	5,26	2,54	2,41	5,12
4	10.35	14,2	14,18	0,14	0,28	0,28	3,57	3,98	3,97	0,25
5	10.40	14,35	14,33	0,14	0,36	0,35	2,78	5,17	5,02	2,90
6	10.45	14,4	14,37	0,21	0,37	0,37	0,00	5,33	5,32	0,19
7	10.50	14,26	14,24	0,14	0,33	0,32	3,03	4,71	4,56	3,18
8	10.55	14,25	14,22	0,21	0,33	0,32	3,03	4,70	4,55	3,19
9	11.00	14,4	14,4	0,00	0,37	0,37	0,00	5,33	5,33	0,00
10	11.05	14,4	14,4	0,00	0,37	0,37	0,00	5,33	5,33	0,00
11	11.10	14,4	14,4	0,00	0,37	0,37	0,00	5,33	5,33	0,00
12	11.15	14,4	14,4	0,00	0,37	0,37	0,00	5,33	5,33	0,00
13	11.20	14,37	14,36	0,07	0,36	0,35	5,56	5,17	5,03	2,71

No	Jam	Tegangan DC (V)		Error (%)	Arus DC (A)		Error (%)	Daya (Watt)		Error (%)
		Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW	
14	11.25	14,27	14,25	0,14	0,33	0,32	3,03	4,71	4,56	3,18
15	11.30	14,1	14,08	0,14	0,31	0,31	3,23	4,37	4,36	0,23
X Solar Cell		14,18	14,16	0,11	0,32	0,315	2,53	4,6	4,49	1,99

Dari data yang diperoleh terjadi perbedaan hasil, antara data yang tertampil di serial monitor dengan *LabVIEW* baik pada pengukuran *output solar cell* maupun baterai. Persentase rata-rata *error* yang dihasilkan pada pengukuran *output solar cell* yakni memperoleh nilai 0,22% untuk tegangan DC, 1,70% untuk arus DC dan 1,90% untuk daya DC. Sedangkan persentase rata-rata *error* yang dihasilkan pada pengukuran *output* baterai yakni memperoleh nilai 0,11% untuk tegangan DC, 2,53% untuk arus DC dan 1,99% untuk daya DC. Nilai rata-rata persentase kesalahan yang diperoleh tergolong baik karena nilainya masih dibawah 5% sesuai dengan SNI. Kemungkinan *error* terjadi karena nilai output dari *solar cell* yang terkadang berubah-ubah (tidak konstan) ataupun bisa dari sensor-sensor yang digunakan.

Kemudian nilai tertinggi tegangan, arus dan daya yang dihasilkan pada *solar cell* maupun baterai terjadi pada pukul 11.00 dengan nilai 15,5 volt, 0,56 ampere dan 8,68 watt untuk *solar cell* dan 14,4 volt, 0,37 ampere dan 5,328 watt untuk baterai. Kemudian nilai terendah tegangan, arus dan daya yang dihasilkan pada *solar cell* maupun baterai terjadi pada pukul 10.25 dengan nilai 13,35 volt, 0,36 ampere dan 4,806 watt untuk *solar cell* dan 13,3 volt, 0,19 ampere dan 2,527 watt untuk baterai. Nilai tegangan berbanding lurus dengan nilai arusnya, semakin besar tegangan yang mengalir maka semakin besar pula arusnya. Kemudian kita dapat menghitung berapa lama baterai akan terisi penuh dengan mengambil nilai arus rata-rata yang mengalir ke baterai yaitu 0.32 ampere menggunakan persamaan (3.4):

$$T_a = \frac{C}{I} (1 + 20\%)$$

$$T_a = \frac{7.2 \text{ Ah}}{0.32 \text{ A}} (1 + 20\%)$$

$$T_a = 22.5 \text{ h} (1 + 20\%)$$

$$T_a = 27 \text{ hours}$$

Waktu yang dibutuhkan *solar cell* untuk mengisi sebuah baterai dengan kapasitas 7,2 Ah dibutuhkan waktu selama 27 jam.

4.7 Pengujian Variasi Kecepatan Blower Terhadap V, I, f dan RPM pada PLTB

Pengujian dilakukan dengan menyambungkan output PLTB dengan beban resistor keramik 120 Ω. Kincir PLTB berputar sesuai dengan kecepatan blower yang diberikan. Blower berfungsi

sebagai sumber angin yang digunakan untuk memutar turbin pada PLTB yang telah dirancang. Digital multimeter difungsikan sebagai pembaca nilai frekuensi yang dihasilkan dari pembangkit. Berikut hasil perolehan datanya yang dapat dilihat pada **Error! Reference source not found.**

Tabel 4.10 Pengujian Variasi Kecepatan Blower Terhadap V, I, f dan RPM pada PLTB

Kecepatan Blower	Tegangan RS (V)		Error (%)	Arus RS (A)		Error (%)	Frekuensi RS (Hz)	RPM Turbin
	Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW			
50	15,63	15,60	0,19	0,11	0,10	9,09	22,3	787
51	16,27	16,24	0,18	0,12	0,11	8,33	23,08	817
52	16,51	16,49	0,12	0,12	0,12	0,00	22,7	825
53	17,53	17,53	0,00	0,13	0,12	7,69	24,18	855
54	17,71	17,67	0,23	0,13	0,13	0,00	24,49	873
55	18,01	17,96	0,28	0,14	0,13	7,14	24,8	899
56	18,25	18,18	0,38	0,14	0,14	0,00	25,38	917
\bar{X} Error (%)			0,20	\bar{X} Error (%)		4,61		
Kecepatan Blower	Tegangan ST (V)		Error (%)	Arus ST (A)		Error (%)	Frekuensi ST (Hz)	RPM Turbin
	Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW			
50	15,61	15,60	0,06	0,11	0,11	0,00	21,7	780
51	16,25	16,25	0,00	0,12	0,11	8,33	22,53	811
52	16,7	16,67	0,18	0,12	0,11	8,33	23,3	834
53	17,23	17,2	0,17	0,12	0,12	0,00	23,63	850
54	17,31	17,27	0,23	0,13	0,12	7,69	24,17	862
55	17,71	17,65	0,34	0,13	0,13	0,00	24,54	898
56	18,13	18,08	0,28	0,13	0,13	0,00	25,05	905
\bar{X} Error (%)			0,18	\bar{X} Error (%)		3,48		
Kecepatan Blower	Tegangan RT (V)		Error (%)	Arus RT (A)		Error (%)	Frekuensi RT (Hz)	RPM Turbin
	Serial Monitor Arduino	LabVIEW		Serial Monitor Arduino	LabVIEW			
50	15,58	15,55	0,19	0,11	0,11	0,00	21,45	778
51	15,84	15,8	0,25	0,12	0,11	8,33	22,06	795
52	16,41	16,36	0,30	0,12	0,11	8,33	22,54	820
53	16,71	16,68	0,18	0,12	0,11	0,00	23,36	844
54	17,34	17,30	0,23	0,13	0,12	7,69	23,91	865
55	17,71	17,69	0,11	0,13	0,13	0,00	24,5	896
56	18,21	18,21	0,00	0,14	0,13	7,14	25,23	910
\bar{X} Error (%)			0,18	\bar{X} Error (%)		4,50		

Dari data yang diperoleh terjadi perbedaan hasil, antara data yang tertampil di serial monitor dengan LabVIEW baik pada kabel RS, kabel ST maupun kabel RT. Persentase rata-rata error yang dihasilkan pada pengujian PLTB kabel RS yakni memperoleh nilai 0,20% untuk tegangan AC dan 4,61% untuk arus AC. Lalu persentase rata-rata error yang dihasilkan pada pengujian PLTB kabel ST yakni memperoleh nilai 0,18% untuk tegangan AC dan 3,48% untuk arus AC. Kemudian persentase rata-rata error yang dihasilkan pada pengujian PLTB kabel RT yakni memperoleh nilai 0,18% untuk tegangan AC dan 4,50% untuk arus AC. Nilai rata-rata persentase kesalahan yang diperoleh tergolong baik karena nilainya masih dibawah 5% sesuai dengan SNI. Kemungkinan error terjadi karena nilai output dari PLTB yang terkadang berubah-ubah (tidak konstan) ataupun bisa dari sensor-sensor yang digunakan.

Nilai kecepatan blower berbanding lurus dengan nilai tegangan, arus, frekuensi serta rpm turbin. Semakin besar nilai kecepatan blowernya maka semakin besar pula nilai tegangan, arus, frekuensi serta rpm turbin. Pada kecepatan blower yang sama, besarnya nilai tegangan pada penghantar RS, ST dan RT tidak jauh berbeda. Hal ini sesuai dengan teori sistem 3 fasa bahwa tegangan yang mengalir pada tiap fasanya seimbang namun memiliki perbedaan fasa yang berbeda sebesar 120 derajat. Kemudian untuk mencari berapa daya yang dihasilkan oleh generator harus dicari V phasanya terlebih dahulu menggunakan persamaan (3.5):

$$V_{R-S} = \sqrt{3} \cdot V_{Line R}$$

$$18,25 = \sqrt{3} \cdot V_{Line}$$

$$V_{Line R} = \frac{18,25}{\sqrt{3}} = 10,53 \text{ volt}$$

Dengan rumus yang sama maka diperoleh tegangan tiap fasa RST. Perhitungan kapasitas daya dihitung berdasarkan nilai tegangan dan arus maksimal yang dihasilkan oleh generator. Tegangan tiap fasa dapat dilihat pada **Error! Reference source not found.**

Tabel 4.11 Tegangan Tiap Fasa

VR-N (V)	VS-N (V)	VT-N (V)
10,53	10,46	10,51

Kemudian arus yang digunakan menggunakan arus tertinggi dari tiap fasanya dimana nilai I fasa = I line sesuai dengan persamaan (3.6). Nilai arus tiap fasa dapat dilihat pada **Error! Reference source not found.**

Tabel 4.12 Arus Tiap Fasa

IR (A)	IS (A)	IT (A)
0,14	0,13	0,14

Dengan demikian dapat dihitung kapasitas daya yang dihasilkan oleh generator menggunakan persamaan (3.7) dengan asumsi $\cos\phi = 1$ karena beban yang digunakan berupa resistor keramik 15 watt 12 ohm.

$$P = (V_{Line R} \cdot I_R \cdot \cos\phi) + (V_{Line S} \cdot I_S \cdot \cos\phi) + (V_{Line T} \cdot I_T \cdot \cos\phi)$$

$$P = (10,53 \cdot 0,14 \cdot 1) + (10,46 \cdot 0,13 \cdot 1) + (10,51 \cdot 0,14 \cdot 1)$$

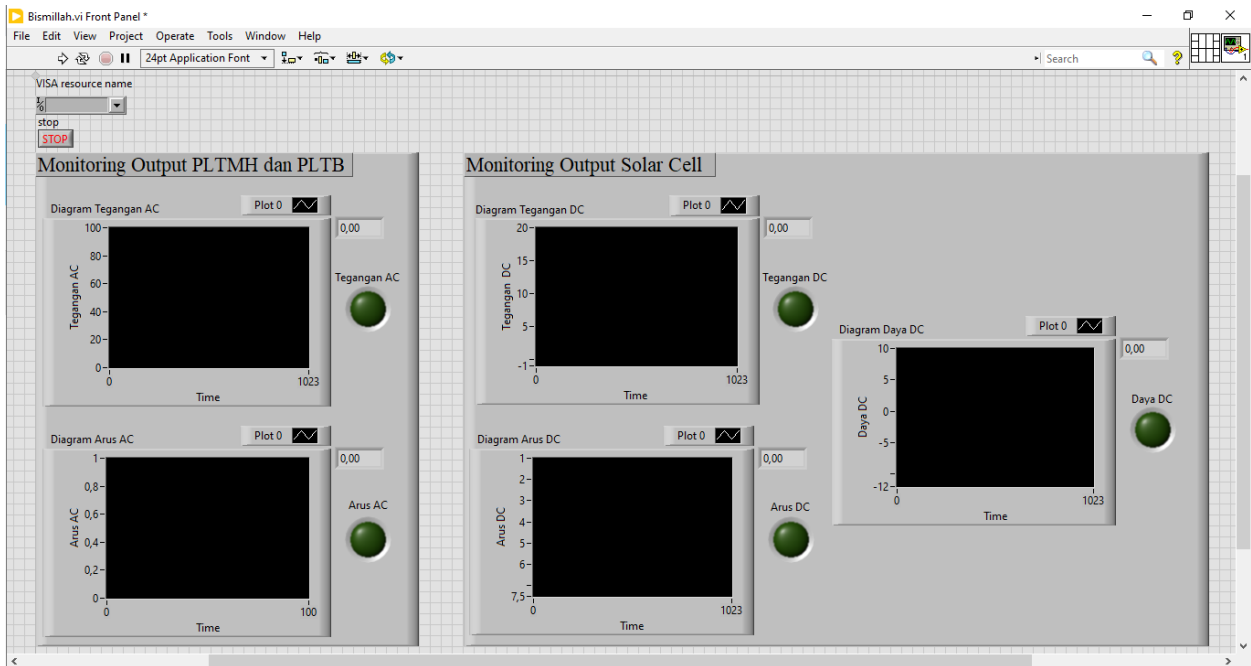
$$P = 1,4742 + 1,3598 + 1,4714$$

$$P = 4,3054 \text{ watt}$$

Jadi kapasitas daya yang dihasilkan oleh PLTB Laboratorium Sistem Ketenagaan Universitas ISLAM Indonesia sebesar 4,3054 watt dengan frekuensi yang tetap berada pada nilai 25 Hz. Nilai frekuensi berbanding lurus dengan kecepatan generator. Semakin besar kecepatan generator berputar maka tegangan dan arus yang dihasilkan pun akan semakin besar dengan nilai beban yang sama.

4.8 Penilaian *User* Terhadap Tampilan dan Fungsi Sistem Monitoring

Penilaian *user* terhadap tampilan dan fungsi sistem monitoring 3 pembangkit hibrida pada laboratorium sistem tenaga universitas ISLAM Indonesia dilakukan dengan cara mengisi kuesioner yang telah dibuat menggunakan *google form* terhadap 5 orang yang berbeda. Berikut tampilan sistem monitoringnya dapat dilihat pada Gambar 4.1 dan hasil penilaian penggunaanya dapat dilihat pada Gambar 4.2.



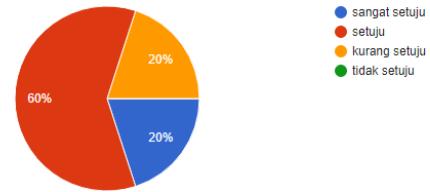
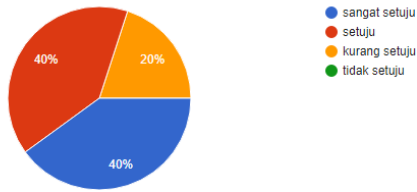
Gambar 4.1 Tampilan Monitoring Front Panel Pada *LabVIEW*

Ketika tegangan *AC*, tegangan *DC*, arus *AC* dan arus *DC* mengalir pada *LabVIEW* maka *LabVIEW* akan menampilkan hasilnya dalam bentuk grafik, angka serta lampu indikator akan menyala dengan warna hijau muda. Sebaliknya ketika parameter-parameter tersebut tidak mengalir pada *LabVIEW* maka *LabVIEW* tidak akan menampilkan data apapun baik berupa grafik, angka serta lampu indikator akan mati ditandai dengan warna hijau tua. Tampilan grafik berfungsi untuk menjabarkan data dalam bentuk naik turunnya data ke 3 pembangkit hibrida agar memudahkan pengguna dalam melakukan analisa. Pembacaan data juga sudah cukup jelas dan mudah dipahami dimana kolom pembacaan output dari PLTMH dan PLTB berada di posisi bagian sebelah kiri dengan nilai tegangan *AC* dan arus *AC*, sedangkan output dari *solar cell* berada di posisi bagian sebelah kanan dengan nilai tegangan *DC*, arus *DC* dan daya *DC*.

Apakah pengguna dapat memahami dengan mudah cara penggunaan alat? Apakah pengguna dapat mengoperasikan alat dengan benar?

5 jawaban

5 jawaban

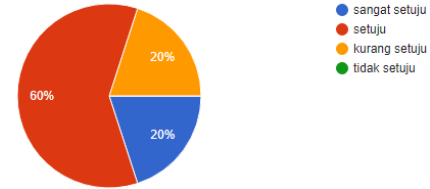
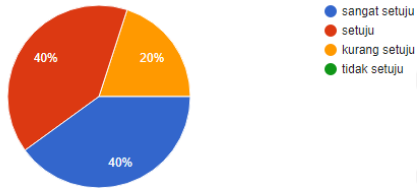


Apakah pengguna dapat membaca tampilan alat dengan mudah?

5 jawaban

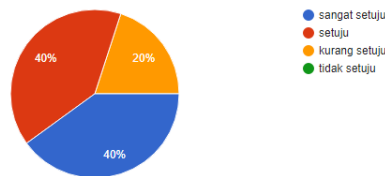
Apakah pengguna dapat memahami tampilan alat dengan benar?

5 jawaban



Apakah alat monitoring cukup membantu pengguna dalam membaca parameter-parameter pembangkit listrik?

5 jawaban



Gambar 4.2 Penilaian User Terhadap Tampilan dan Fungsi Sistem Monitoring

Penilaian *user* pun berbeda-beda untuk pertanyaan pertama yaitu “Apakah pengguna dapat memahami dengan mudah cara penggunaan alat?” didapatkan hasil 40% sangat setuju, 40% setuju dan 20% kurang setuju. Pertanyaan kedua “Apakah pengguna dapat mengoperasikan alat dengan benar?” didapatkan hasil 20% sangat setuju, 60% setuju dan 20% kurang setuju. Pertanyaan ketiga “Apakah pengguna dapat membaca tampilan alat dengan mudah?” didapatkan hasil 40% sangat setuju, 40% setuju dan 20% kurang setuju. Pertanyaan keempat “Apakah pengguna dapat membaca tampilan alat dengan benar?” didapatkan hasil 20% sangat setuju, 60% setuju dan 20% kurang setuju. Pertanyaan terakhir “Apakah alat monitoring cukup membantu pengguna dalam membaca parameter-parameter pembangkit listrik?” didapatkan hasil 40% sangat setuju, 40% setuju dan 20% kurang setuju. Kemudian untuk saran yang diterima ada 2 yaitu:

1. Tampilan monitoring pada layar mungkin bisa lebih dibuat lebih menarik lagi agar pengguna bisa lebih nyaman dan mudah membaca tiap-tiap parameter yang ada.
2. Besarkan tampilan indikator tegangan dan arus agar lebih mudah terbaca dan tidak kalah besar dari grafiknya

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Sistem monitoring yang telah dirancang berfungsi dengan baik dan mudah dipahami.
2. Persentase keberhasilan pembacaan sensor yang telah dirancang memperoleh nilai sebesar 98,33% untuk zmpt101b, 99,94% untuk voltage sensor, 92,19% untuk acs712 arus AC dan 92,94% untuk acs712 arus DC.
3. Persentase rata-rata *error* data antara yang tertampil di serial monitor dengan *LabVIEW* pada pengujian PLTMH, *Solar Cell* dan PLTB sudah termasuk golongan alat SNI dimana nilai yang didapatkan dibawah 5%.

5.2 Saran

1. Diharapkan sistem monitoring selanjutnya menggunakan sistem tanpa kabel (*IoT*) agar lebih efisien waktu dan tenaga.
2. Diharapkan komunikasi dari mikrokontroler terhadap sensor dapat menggunakan papan pcb agar komponen lebih rapi, tertata, efisien dan lebih kuat. Kesalahan pembacaan sensor sering terjadi karena ada kabel *male* ataupun *female* yang longgar / tidak kuat / tidak menancap pada komponen.

DAFTAR PUSTAKA

- [1] A. Yulistiyanto, "SISTEM MONITORING ENERGI LISTRIK PADA MOTOR LISTRIK OVEN 7 PT. RECIKTT BENCKISER MENGGUNAKAN IoT (INTERNET of THINGS)," 2019, [Online]. Available: <http://repository.unissula.ac.id/id/eprint/14767>.
- [2] V. S. Sathishkumar, N. Ranjitha, A. Sabareesh, S. Sandhiya, and A. Sandhiya, "LABVIEW based thermal power plant Monitoring and controlling using GSM technology," no. 1.
- [3] A. H. K. L. U. Abdurrahman, "Sistem Monitoring Output Solar Panel Menggunakan Labview," *Epic*, vol. 3, no. 1, pp. 1–6, 2020, doi: 10.32493/epic.v3i1.3796.
- [4] R. Sandra, V. Simbar, and A. Syahrin, "Prototype Sistem Pendeteksi Darah Menggunakan Arduino Uno R3," vol. 8, no. 1, pp. 80–86, 2017.
- [5] M. Y. Haris, "Perancangan Sistem Kontrol Lampu Berbasis Mikrokontroler Arduino Uno R3 Dengan Sensor Suara," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2017.
- [6] Nusa dkk., "Sistem Monitoring Konsumsi Energi Listrik Secara Real Time Berbasis Mikrokontroler," *E-Jurnal Tek. Elektro dan Komput.*, vol. 4, no. 5, pp. 19–26, 2015.
- [7] A. A. Maulana, "Prototype Pengendali Lampu Berbasis Myrio Dan Labview," pp. 1–20, 2018.
- [8] U. S. Utara, U. S. Utara, and U. S. Utara, "Rancangan Alat Ukur Daya Listrik Digital Berbasis Arduino," 2019.
- [9] A. Julisman, I. D. Sara, and R. H. Siregar, "Prototipe Pemanfaatan Panel Surya Sebagai Sumber Energi Pada Sistem Otomasi Stadion Bola," *Kitektro*, vol. 2, no. 1, pp. 35–42, 2017.

LAMPIRAN

Lampiran 1 – Spesifikasi Voltage Sensor

Spesifikasi Voltage Sensor	
Tegangan masukan	0 V – 25 V (DC)
Output interface	positif koneksi 5/3.3 V negatif koneksi ke GND huruf “s” terhubung ke arduino pin A0
Tegangan analog	0,00489 V
Tegangan DC masukan antarmuka	terminal positif ke VCC terminal negatif ke GND
Jangkauan tegangan yang dideteksi	0.02445 V – 25 V (DC)
DC antar masukan	red terminal positif ke VCC negatif ke GND

Lampiran 2 – Spesifikasi ZMPT101B

Spesifikasi Elektrik ZMPT101B	
Range Frekuensi	50-60 Hz
Resistansi DC 20°C	110 Ω
Nilai Beban	≤ 200 Ω
Arus Primer	2 mA
Arus Sekunder	2 mA
Rasio Balik	1000:1000
Linearitas	0.1 %
Jarak Arus	0-3 mA
Tingkat Akurasi	0.2

Lampiran 3 – Penjelasan Skematik ACS712

No	Nama	Penjelasan
1 dan 2	IP+	Pin mendeteksi arus
3 dan 4	IP-	Pin mendeteksi arus
5	GND	Pin ground
6	FILTER	Pin untuk kapasitor eksternal yang digunakan menentukan bandwidth
7	VOUT	Arus output yang dihitung
8	VCC	Tegangan power supply

Lampiran 4 – Penjelasan Spesifikasi Arduino Mega

Penjelasan Arduino Mega	
Mikrokontroller	ATmega2560
Pin Analog	16
Digital I/O	54 pin (14 pin untuk PWM)
Operasi Tegangan	5 V
Input Tegangan	7-12 V
Input Tegangan batas	6-20 V
Arus DC untuk pin 3.3V	50 mA
Arus DC Tiap Pin I/O	20 mA

Penjelasan Arduino Mega	
Memori Flash	256 KB
Bootloader	SRAM 8 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Lampiran 5 – Program Arduino

```

unsigned int waveHolderValue[20];
unsigned int maxValueHolder[50];
byte frequencyStep = 0;
byte roll = 0;
byte sensorCode = 0;
const byte totalDataSample[4] = { 11, 25, 11, 21 };
const unsigned int lowestSampleValue[4] = { 520, 512, 343, 512 };
const unsigned int highestSampleValue[4] = { 600, 536, 628, 532 };
const byte frequency = 50;
const byte totalWaveValue = 20;
float acVoltageSampleData[11][2] = { { 520, 15 }, { 530, 20 }, { 540, 25 }, { 550, 30 }, { 555, 40 }, { 560, 50 }, { 565, 60 }, { 570, 70 },
{ 580, 80 }, { 590, 90 }, { 600, 100 } };
float acCurrentSampleData[25][2] = { { 512, 0 }, { 513, 0.05 }, { 514, 0.09 }, { 515, 0.13 }, { 516, 0.17 }, { 517, 0.21 }, { 518, 0.25 }, { 519,
0.29 }, { 520, 0.33 }, { 521, 0.37 }, { 522, 0.41 }, { 523, 0.45 },
{ 524, 0.49 }, { 525, 0.53 }, { 526, 0.57 }, { 527, 0.61 }, { 528, 0.65 }, { 529, 0.69 }, { 530, 0.73 }, { 531, 0.77 }, { 532, 0.81 }, { 533, 0.85 },
{ 534, 0.89 }, { 535, 0.93 }, { 536, 0.97 } };
float dcVoltageSampleData[11][2] = { { 343, 8.40 }, { 541, 13.32 }, { 543, 13.37 }, { 554, 13.65 }, { 578, 14.25 }, { 581, 14.32 }, { 582,
14.35 }, { 585, 14.42 }, { 593, 14.62 }, { 612, 15.1 }, { 628, 15.5 } };
float dcCurrentSampleData[21][2] = { { 511, 0.11 }, { 513, 0.05 }, { 514, 0.1 }, { 515, 0.15 }, { 516, 0.2 }, { 517, 0.25 }, { 518, 0.3 }, { 519,
0.35 }, { 520, 0.4 }, { 521, 0.45 }, { 522, 0.5 }, { 523, 0.55 },
{ 524, 0.6 }, { 525, 0.65 }, { 526, 0.7 }, { 527, 0.75 }, { 528, 0.8 }, { 529, 0.85 }, { 530, 0.9 }, { 531, 0.95 }, { 532, 1 } };
float linearRegressionValue[2];
float powerSource[4];

void setup() {
  Serial.begin(9600); // set data rate in bits per second
  // Serial.println("System initialization ready"); // show text if system is ready to init
  pinMode(A0, INPUT); // AC Voltage sensor
  pinMode(A1, INPUT); // AC Current sensor
  pinMode(A2, INPUT); // DC Voltage sensor
  pinMode(A3, INPUT); // DC Current sensor
  // Serial.println("Initialization done & program ready to run"); // Show text if system init done
}

void loop() {
  unsigned int ADCInputValue;
  switch(sensorCode) {

```

```

case 0: ADCInputValue = analogRead(A0); break;
case 1: ADCInputValue = analogRead(A1); break;
case 2: ADCInputValue = analogRead(A2); break;
case 3: ADCInputValue = analogRead(A3); break;
}

if(roll < totalWaveValue) {
    waveHolderValue[roll] = ADCInputValue;
    roll++;

} else {
    roll = 0;

    if(frequencyStep < frequency) {
        maxValueHolder[frequencyStep] = getMaxValue();
        frequencyStep++;

    } else {
        frequencyStep = 0;

        switch(sensorCode) {
//      case 0: showValue("Nilai input ADC AC Voltage adalah : ", "Nilai tegangan AC adalah : "); break;
case 0: showValue("A"); break;
//      case 1: showValue("Nilai input ADC AC Current adalah : ", "Nilai arus AC adalah : "); break;
case 1: showValue("B"); break;
//      case 2: showValue("Nilai input ADC DC Voltage adalah : ", "Nilai tegangan DC adalah : "); break;
case 2: showValue("C"); break;
//      case 3: showValue("Nilai input ADC DC Current adalah : ", "Nilai arus DC adalah : "); break;
case 3: showValue("D"); break;
        }

    }
}

delayMicroseconds(1000);
}

//void showValue(String text1, String text2) {
void showValue(String text0) {
    unsigned int adcFinalValue = getAverage();
    Serial.print(text0);
//    Serial.println(adcFinalValue);
    float finalValue = getFinalValue(adcFinalValue);
//    Serial.print(text2);

```

```

Serial.println(finalValue);

switch(sensorCode) {
  case 0: sensorCode = 1; powerSource[0] = finalValue; break;
  case 1: sensorCode = 2; powerSource[1] = finalValue; break;
  case 2: sensorCode = 3; powerSource[2] = finalValue; break;
  case 3: sensorCode = 0; powerSource[3] = finalValue; break;
}

if(sensorCode == 0) {
  float valuePower;
  // valuePower = powerSource[0] * powerSource[1];
  // Serial.print("Nilai daya AC adalah : ");
  // Serial.println(valuePower);
  valuePower = powerSource[2] * powerSource[3];
  // Serial.print("Nilai daya DC adalah : ");
  Serial.print("E");
  Serial.println(valuePower);
  // Serial.println();
}
}

unsigned int getMaxValue() {

  unsigned int maxValue = 0;
  for(byte i = 0; i < totalWaveValue; i++) {

    if(maxValue < waveHolderValue[i]) {
      maxValue = waveHolderValue[i];
    }
  }

  return maxValue;
}

// get average value from all wave highest value
unsigned int getAverage() {

  unsigned int valueHolder = 0;
  for (byte i = 0; i < frequency; i++) {

    valueHolder += maxVaholder[i];
  }
}

```



```

valueHolder = valueHolder / frequency;

return valueHolder;
}

// Final process to get final output value from ADC Input
float getFinalValue(unsigned int adcFinalValue) {
    float voltageValue;
    unsigned int lowestADCInputSample;
    unsigned int highestADCInputSample;
    byte mTotalDataSample;

    switch(sensorCode) {
        case 0: lowestADCInputSample = lowestSampleValue[0]; highestADCInputSample = highestSampleValue[0];
mTotalDataSample = totalDataSample[0]; break;
        case 1: lowestADCInputSample = lowestSampleValue[1]; highestADCInputSample = highestSampleValue[1];
mTotalDataSample = totalDataSample[1]; break;
        case 2: lowestADCInputSample = lowestSampleValue[2]; highestADCInputSample = highestSampleValue[2];
mTotalDataSample = totalDataSample[2]; break;
        case 3: lowestADCInputSample = lowestSampleValue[3]; highestADCInputSample = highestSampleValue[3];
mTotalDataSample = totalDataSample[3]; break;
    }

    if(adcFinalValue < lowestADCInputSample) {

        voltageValue = 0;

    } else if(adcFinalValue > highestADCInputSample) {
        voltageValue = 0;

    } else if(adcFinalValue == lowestADCInputSample) {

        voltageValue = 0;

    } else {

        for(unsigned int i = 0; i < mTotalDataSample; i++) {
            if(sensorCode == 0) {
                if((adcFinalValue > acVoltageSampleData[i][0]) && (adcFinalValue <= acVoltageSampleData[i+1][0])) {
                    voltageValue = ((adcFinalValue - acVoltageSampleData[i][0]) * ((acVoltageSampleData[i+1][1] -
acVoltageSampleData[i][1]) /
                    (acVoltageSampleData[i+1][0] - acVoltageSampleData[i][0]))) + acVoltageSampleData[i][1];
                    break;
                }
            }
        }
    } else if(sensorCode == 1) {

```

```

        if((adcFinalValue > acCurrentSampleData[i][0]) && (adcFinalValue <= acCurrentSampleData[i+1][0])) {
            voltageValue = ((adcFinalValue - acCurrentSampleData[i][0]) * ((acCurrentSampleData[i+1][1] -
acCurrentSampleData[i][1]) /
            (acCurrentSampleData[i+1][0] - acCurrentSampleData[i][0]))) + acCurrentSampleData[i][1];
            break;
        }
    } else if(sensorCode == 2) {
        if((adcFinalValue > dcVoltageSampleData[i][0]) && (adcFinalValue <= dcVoltageSampleData[i+1][0])) {
            voltageValue = ((adcFinalValue - dcVoltageSampleData[i][0]) * ((dcVoltageSampleData[i+1][1] -
dcVoltageSampleData[i][1]) /
            (dcVoltageSampleData[i+1][0] - dcVoltageSampleData[i][0]))) + dcVoltageSampleData[i][1];
            break;
        }
    } else {
        if((adcFinalValue > dcCurrentSampleData[i][0]) && (adcFinalValue <= dcCurrentSampleData[i+1][0])) {
            voltageValue = ((adcFinalValue - dcCurrentSampleData[i][0]) * ((dcCurrentSampleData[i+1][1] -
dcCurrentSampleData[i][1]) /
            (dcCurrentSampleData[i+1][0] - dcCurrentSampleData[i][0]))) + dcCurrentSampleData[i][1];
            break;
        }
    }
}
}
return voltageValue;
}

```



الجامعة الإسلامية
 الباندا انديونيسيا