

**RANCANG BANGUN *PROTOTYPE* MONITORING
KETIDAKSEIMBANGAN BEBAN PADA TRANSFORMATOR
DISTRIBUSI BERBASIS *INTERNET OF THINGS***

SKRIPSI

OLEH:

**ZULFITRA
168120031**



**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2022**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 20/6/22

Access From (repository.uma.ac.id)20/6/22

RANCANG BANGUN *PROTOTYPE* MONITORING KETIDAKSEIMBANGAN BEBAN PADA TRANSFORMATOR DISTRIBUSI BERBASIS *INTERNET OF THINGS*

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana di Fakultas Teknik
Universitas Medan Area



Oleh:

**ZULFITRA
168120031**

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2022**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

Document Accepted 20/6/22

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Access From (repository.uma.ac.id)20/6/22

HALAMAN PENGESAHAN

Judul Skripsi : Rancang Bangun *Prototype* Monitoring Ketidakseimbangan Beban pada Transformator Distribusi Berbasis *Internet of Things*

Nama : Zulfitra

NPM : 16.812.0031

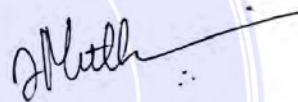
Fakultas : Teknik

Disetujui Oleh
Komisi Pembimbing



Prof. Dr. Dadan Ramdan, M.Eng, M.Sc

Pembimbing I



Syarifah Muthia Putri, ST, MT

Pembimbing II



Ge Ralman Syah, S.Kom, M.Kom

Dekan



Pratiwi S.Pd, MT

Ka. Prodi

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila dikemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 17 Februari 2022



METERAI
TEMPEL
55AEAJX630228706

Zulfitra
16.812.0031

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR/SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Medan Area, saya yang bertanda tangan dibawah ini:

Nama : Zulfitra

NPM : 16.812.0031

Program Studi : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Tugas Akhir/Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti NonEksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

"Rancang Bangun *Prototype* Monitoring Ketidakseimbangan Beban pada Transformator Distribusi Berbasis *Internet of Things*". Dengan Hak Bebas Royalti Noneklusif ini, Universitas Medan Area berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir/skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Medan, 17 Februari 2022

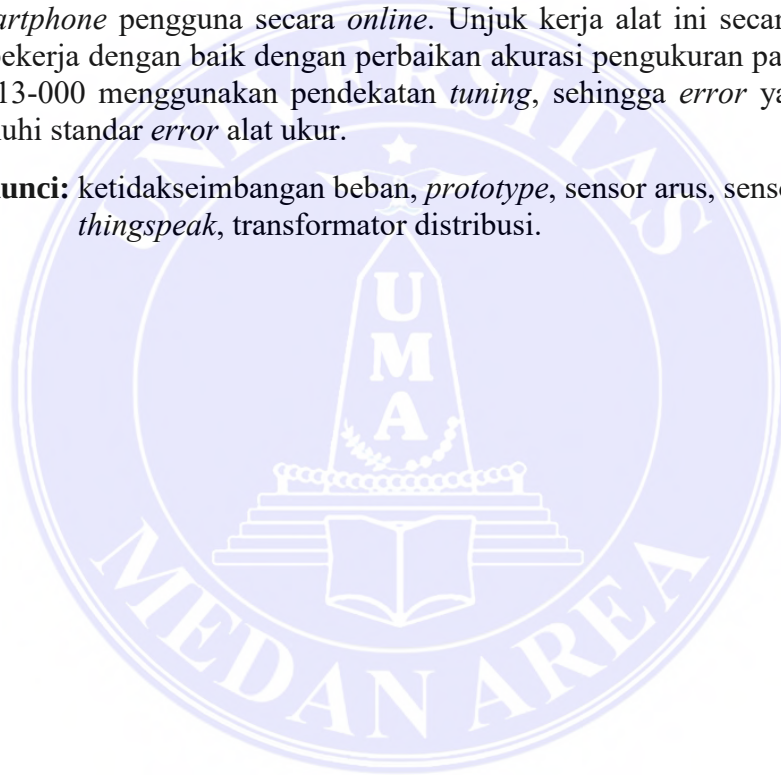


Zulfitra

ABSTRAK

Ketidakseimbangan beban dengan selisih yang cukup jauh dalam sistem distribusi tenaga listrik menyebabkan munculnya arus netral, sehingga menimbulkan *losses* akibat mengalirnya arus netral ke tanah, panas berlebih pada salah satu fasa dan mengakibatkan penurunan kualitas serta kerusakan pada transformator distribusi bila berlangsung dalam waktu yang lama. Maksud penelitian ini adalah merancang dan membuat suatu prototipe monitoring pengukuran arus, tegangan dan persentase ketidakseimbangan beban pada *prototype* transformator distribusi, dengan tujuan memberikan informasi pengukuran dari *prototype* melalui internet. Arduino Uno sebagai pengolah data dari sensor arus dan sensor tegangan, serta ESP8266-01 sebagai modul *WiFi* yang berfungsi dalam mengirimkan data pengukuran arus setiap fasa, tegangan *line to neutral* serta persentase ketidakseimbangan beban ke *Web server Thingspeak* berupa tujuh tampilan grafik yang dapat diakses oleh PC/*smartphone* pengguna secara *online*. Unjuk kerja alat ini secara keseluruhan dapat bekerja dengan baik dengan perbaikan akurasi pengukuran pada sensor arus SCT 013-000 menggunakan pendekatan *tuning*, sehingga *error* yang dihasilkan memenuhi standar *error* alat ukur.

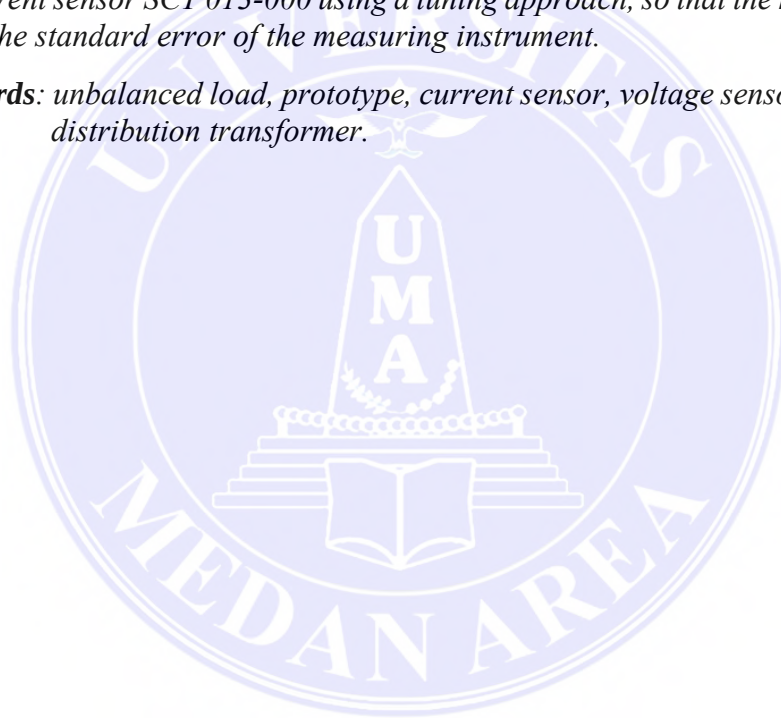
Kata kunci: ketidakseimbangan beban, *prototype*, sensor arus, sensor tegangan, *thingspeak*, transformator distribusi.



ABSTRACT

The Unbalanced load with a significant difference in the electricity distribution system causes the emergence of a neutral current, causing losses due to the flow of neutral current to the ground, overheating in one phase, and resulting in a decrease in quality and damage to the distribution transformer if it lasts for an extended period of time. The goal of this research is to design and build a monitoring prototype for measuring current, voltage, and percentage of unbalance on a distribution transformer prototype, with the goal of providing measurement data from the prototype via the internet. The Arduino Uno serves as a data processor from current and voltage sensors, and the ESP8266-01 serves as a wifi module, sending current measurement data for each phase, line to neutral voltage, and the percentage of load imbalance to the Thingspeak Web Server in the form of seven graphic displays that PC/ smartphone users can access online. The overall performance of this tool can work well by improving the measurement accuracy of the current sensor SCT 013-000 using a tuning approach, so that the resulting error meets the standard error of the measuring instrument.

Keywords: *unbalanced load, prototype, current sensor, voltage sensor, thingspeak, distribution transformer.*

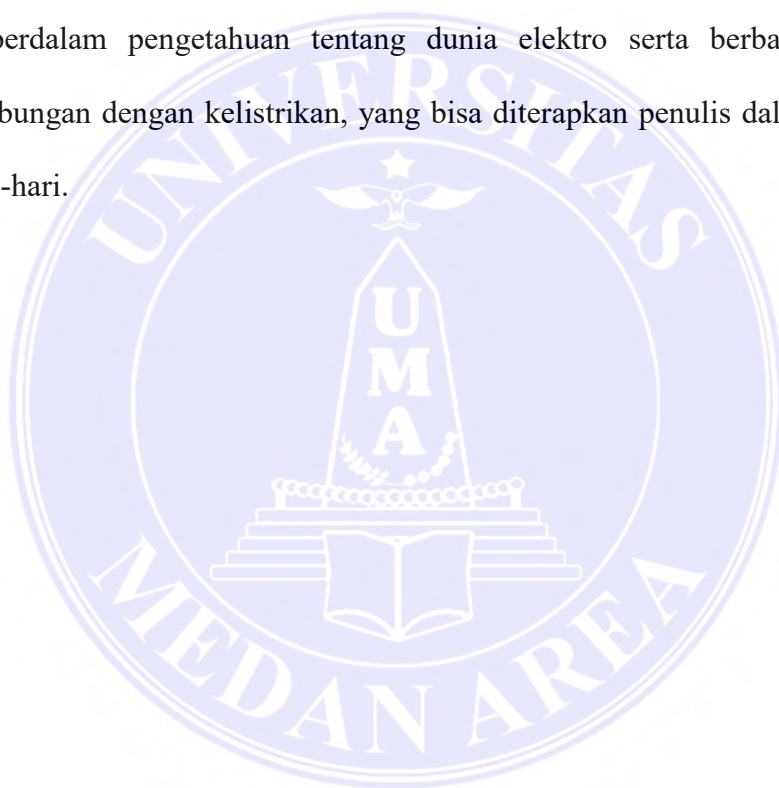


RIWAYAT HIDUP

Penulis dilahirkan di Panipahan pada tanggal 07 Oktober 1997 dari ayah Alm. Nordin dan ibu Zaitun. Penulis merupakan anak kelima dari lima bersaudara.

Tahun 2016 penulis lulus dari Madrasah Aliyah Ishlahiyah Panipahan dan pada tahun 2016 terdaftar sebagai mahasiswa Fakultas Teknik Universitas Medan Area Jurusan Teknik Elektro.

Selama mengikuti perkuliahan, penulis menjadi lebih memahami serta memperdalam pengetahuan tentang dunia elektro serta berbagai hal yang berhubungan dengan kelistrikan, yang bisa diterapkan penulis dalam kehidupan sehari-hari.



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadiran Allah SWT, yang mana atas rahmat dan hidayah-Nya penulis telah diberikan kesehatan lahir dan batin sehingga dapat menyusun dan menyelesaikan Skripsi ini. Penulis juga menghantarkan shalawat dan salam kepada junjungan kita Nabi Muhammad Shallallahu'alaihi Wasallam, yang telah membawa umat manusia dari masa kebodohan ke masa yang berilmu pengetahuan. Tema yang dipilih dalam penelitian ini adalah “Rancang Bangun *Prototype* Monitoring Ketidakseimbangan Beban pada Transformator Distribusi Berbasis *Internet of Things*”.

Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Sarjana Strata Satu pada program studi Teknik Elektro, Universitas Medan Area.

Selama masa kuliah hingga penyelesaian Skripsi ini, penulis juga banyak mendapat dukungan, bimbingan, maupun bantuan dari berbagai pihak. Untuk itu penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Orang Tua dan segenap keluarga penulis yang telah memberi dukungan baik secara moril maupun materil sehingga Skripsi ini dapat terlaksana.
2. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc, selaku Rektor Universitas Medan Area dan sekaligus Dosen Pembimbing I, yang telah banyak meluangkan waktu, tenaga dan pikiran untuk memberikan bimbingan dan pengarahan kepada penulis hingga penyusunan Skripsi ini selesai.
3. Bapak Dr. Rahmad Syah, S.Kom, M.Kom, selaku Dekan Fakultas Teknik Universitas Medan Area.

4. Bapak Habib Satria, S.Pd, MT, selaku Ketua Program Studi Teknik Elektro Universitas Medan Area.
5. Ibu Syarifah Muthia Putri, ST, MT, selaku Dosen Pembimbing II, yang telah banyak meluangkan waktu dan pikiran untuk memberikan bimbingan dan pengarahan kepada penulis selama perkuliahan hingga penyusunan Skripsi ini selesai.
6. Seluruh Dosen pengajar Universitas Medan Area khususnya Program Studi Teknik Elektro;
7. Seluruh Staff Universitas Medan Area khususnya Fakultas Teknik yang membantu penulis dalam pengurusan administrasi;
8. Seluruh teman-teman Program Studi Teknik Elektro angkatan 2016 atas kerjasama dan kebersamaannya selama menjalani studi;
9. Semua pihak yang tidak bisa penulis sebutkan satu per satu atas bantuan, bimbingan, kritik dan saran.

Penulis menyadari di dalam penulisan Skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan masukan berupa kritik dan saran yang sifatnya membangun demi kesempurnaan Skripsi ini, sehingga kekurangan tersebut tidak terulang lagi pada masa yang akan datang.

Akhirnya penulis berharap semoga Skripsi ini dapat bermanfaat baik untuk kalangan pendidikan maupun masyarakat.

Medan, 17 Februari 2022

Zulfitra

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN	iii
LEMBAR PERNYATAAN	iv
ABSTRAK	v
ABSTRACT.....	vi
RIWAYAT HIDUP.....	vii
KATA PENGANTAR	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
 BAB I PENDAHULUAN	 1
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	3
1.5. Manfaat Penelitian	3
 BAB II TINJAUAN PUSTAKA.....	 4
2.1. Transformator Distribusi.....	4
2.2. Ketidakseimbangan Beban.....	4
2.3. <i>Internet of Things</i>	6
2.4. Sensor YHDC SCT 013-000.....	7
2.5. Sensor Tegangan ZMPT 101B.....	8
2.6. <i>Thingspeak</i>	9
2.7. Arduino UNO.....	10
2.8. ESP8266-01	11
2.9. Manajemen <i>Bandwidth</i>	11
 BAB III METODOLOGI PENELITIAN.....	 13
3.1. Waktu dan Tempat Penelitian	13
3.1.1. Waktu Penelitian	13
3.1.2. Tempat Penelitian	13
3.2. Prosedur Kerja	13
3.2.1. Analisis Kebutuhan Alat dan Bahan.....	14
3.2.2. Menentukan Sensor Arus.....	15
3.2.3. Menentukan Sensor Tegangan.....	18
3.2.4. Menentukan Modul <i>WiFi</i>	19
3.2.5. Pembuatan Program.....	20
3.2.5.1. Menentukan Inisialisasi Port Sensor Arus dan Sensor Tegangan	21
3.2.5.2. Menentukan <i>Channel Thingspeak</i>	23
3.2.6. <i>Compile</i> Program	28
3.2.7. Menentukan Rangkaian Prototipe Transformator Distribusi	29
3.2.8. Menentukan Rangkaian Simulasi Beban 3 Fasa.....	30

3.2.9. Pengaturan Tegangan Eksitasi Generator	31
3.2.10. Diagram Keseluruhan Alat	31
3.2.11. Rangkaian Keseluruhan	33
BAB IV HASIL DAN PEMBAHASAN	35
4.1. Kebutuhan Spesifikasi Minimum <i>Hardware</i> dan <i>Software</i>	35
4.1.1. Rangkaian Mikrokontroler Arduino Uno	35
4.1.2. Rangkaian Sensor Arus SCT 013-000	36
4.1.3. Rangkaian Modul <i>WiFi</i> ESP8266-01	36
4.1.4. Rangkaian Sensor Tegangan ZMPT 101B	37
4.1.5. Rangkaian Keseluruhan	38
4.1.6. Rangkaian Transformator 3 Fasa Hubungan Wye.....	38
4.1.7. Rangkaian Simulasi Beban 3 Fasa.....	40
4.2. Hasil Perancangan Perangkat Lunak (<i>Software</i>).....	41
4.2.1. Hasil Perancangan <i>Channel Thingspeak</i>	41
4.3. Pengujian <i>Prototype</i>	49
4.3.1. Pengujian Karakteristik Sensor Arus SCT 013-000	51
4.3.2. Pengujian Karakteristik Sensor Tegangan ZMPT 101B....	54
4.3.3. Perbaikan Akurasi Pengukuran Sensor Arus Fasa R, Fasa S dan Fasa T dalam Pemrograman Arduino UNO....	58
4.3.4. Pengujian Pengiriman Data <i>Prototype</i> ke <i>Thingspeak</i>	61
BAB V KESIMPULAN DAN SARAN.....	63
5.1. Kesimpulan	63
5.2. Saran	64
DAFTAR PUSTAKA	65
LAMPIRAN.....	67

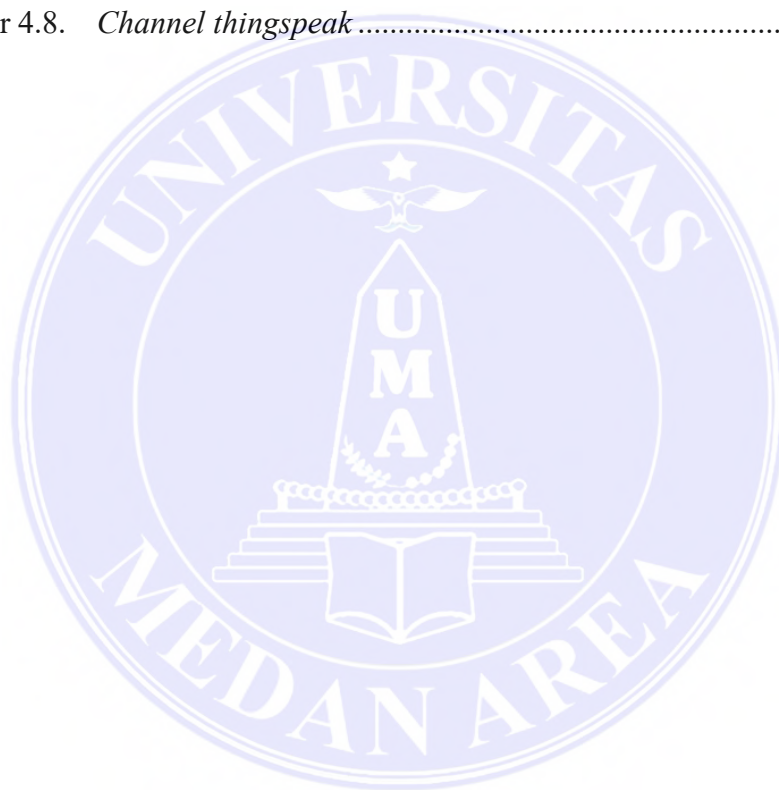
DAFTAR TABEL

	Halaman
Tabel 3.1. Jadwal pelaksanaan kegiatan	13
Tabel 3.2. Komponen di laboratorium yang digunakan	15
Tabel 3.3. Komponen yang penulis gunakan	15
Tabel 3.4. Konfigurasi ESP8266-01 ke Arduino Uno dan FTDI RS232	19
Tabel 4.1. Hasil pengukuran sensor SCT 013-000, sensor ZMPT 101B dan multimeter digital	50
Tabel 4.2. Hasil perbandingan pengukuran arus fasa R sensor SCT 013-000 dan alat ukur Multimeter Digital	51
Tabel 4.3. Hasil perbandingan pengukuran arus fasa S sensor SCT 013-000 dan alat ukur multimeter digital	52
Tabel 4.4. Hasil perbandingan pengukuran arus fasa T sensor SCT 013-000 dan alat ukur multimeter digital	53
Tabel 4.5. Hasil perbandingan perhitungan persentase ketidakseimbangan beban <i>prototype</i> dengan perhitungan secara matematis	54
Tabel 4.6. Hasil perbandingan pengukuran tegangan <i>line to neutral</i> R-N sensor ZMPT 101B dan alat ukur multimeter digital	55
Tabel 4.7. Hasil perbandingan pengukuran tegangan <i>line to neutral</i> S-N sensor ZMPT 101B dan alat ukur multimeter digital	56
Tabel 4.8. Hasil perbandingan pengukuran tegangan <i>line to neutral</i> T-N sensor ZMPT 101B dan alat ukur multimeter digital	57
Tabel 4.9. Perhitungan nilai pendekatan untuk akurasi pengukuran sensor arus fasa R dalam pemrograman arduino	59
Tabel 4.10. Perhitungan nilai pendekatan untuk akurasi pengukuran sensor arus fasa S dalam pemrograman arduino	60
Tabel 4.11. Perhitungan nilai pendekatan untuk akurasi pengukuran sensor arus fasa T dalam pemrograman arduino	60
Tabel 4.12. Pengujian kecepatan transfer data <i>prototype</i> ke <i>Thingspeak</i>	62

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Vektor diagram arus seimbang dan vektor arus tidak seimbang	5
Gambar 2.2. Sensor arus SCT 013-000	7
Gambar 2.3. Sensor tegangan ZMPT 101B	8
Gambar 2.4. <i>Thingspeak</i>	9
Gambar 2.5. Arduino uno.....	10
Gambar 2.6. Modul Wi-Fi ESP8266-01.....	11
Gambar 3.1. Diagram alir penelitian	14
Gambar 3.2. Rangkaian sensor arus menggunakan sensor arus SCT 013-000 pada fasa R, fasa S dan fasa T	16
Gambar 3.3. Skema Pengkondisi sinyal sensor arus	16
Gambar 3.4. Rangkaian sensor tegangan fasa R, fasa S dan fasa T	18
Gambar 3.5. Rangkaian modul <i>WiFi</i> ESP8266-01.....	19
Gambar 3.6. <i>Flowchart</i> perancangan perangkat lunak.....	20
Gambar 3.7. Inisialisasi <i>port</i> sensor arus pada <i>software</i> arduino IDE	21
Gambar 3.8. Inisialisasi <i>port</i> sensor tegangan pada <i>software</i> arduino IDE	22
Gambar 3.9. Halaman login <i>thingspeak</i>	23
Gambar 3.10. Halaman verifikasi email.....	24
Gambar 3.11. Verifikasi akun MathWorks	24
Gambar 3.12. Halaman pembuatan password akun <i>thingspeak</i>	25
Gambar 3.13. Halaman pembuatan akun <i>thingspeak</i> sukses.....	25
Gambar 3.14. Opsi status pengguna akun <i>thingspeak</i>	26
Gambar 3.15. Halaman pembuatan <i>channel</i> baru	26
Gambar 3.16. Halaman <i>form</i> identitas <i>channel</i> baru.....	27
Gambar 3.17. Tampilan status <i>channel</i>	27
Gambar 3.18. <i>Channel</i> ID	28
Gambar 3.19. <i>Write API Key channel prototype</i> monitoring.....	28
Gambar 3.20. <i>Compile</i> program.....	29
Gambar 3.21. Rangkaian 3 fasa hubungan wye dengan beban.....	30
Gambar 3.22. Perangkat simulasi beban	31

Gambar 3.23. Blok diagram perancangan sistem.....	32
Gambar 3.24. Rangkaian keseluruhan sistem	34
Gambar 4.1. Rangkaian arduino uno.....	35
Gambar 4.2. Rangkaian pengkondisi sinyal SCT 013-000	36
Gambar 4.3. Rangkaian modul <i>WiFi</i> ESP8266-01.....	37
Gambar 4.4. Rangkaian sensor tegangan ZMPT 101B.....	37
Gambar 4.5. Hasil perancangan alat.....	38
Gambar 4.6. Rangkaian transformator 3 fasa hubungan wye	39
Gambar 4.7. Rangkaian simulasi beban 3 fasa.....	40
Gambar 4.8. <i>Channel thingspeak</i>	42



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Suatu sistem tenaga listrik terdiri atas sistem pembangkitan, transmisi dan distribusi. Jaringan distribusi merupakan jaringan dalam sistem tenaga listrik yang berhubungan langsung dengan konsumen atau beban. Untuk konsumen berupa rumah tangga menggunakan sisi sekunder dalam jaringan distribusi atau Jaringan Tegangan Rendah (JTR).

Dalam sistem distribusi pada jaringan tegangan rendah umumnya menggunakan transformator distribusi dengan belitan hubungan Bintang/Wye pada sisi sekundernya, dan arus setiap fasa yang melayani beban bernilai sama serta arus pada netral bernilai nol secara teoritis. Dalam prakteknya, arus setiap fasa tidak memiliki nilai yang sama dan arus netral tidak bernilai nol dikarenakan terjadinya pembagian beban-beban yang tidak serempak waktu penyalaan beban-beban tersebut dan penyambungan beban yang dilakukan di saluran distribusi sekunder yang tidak memperhitungkan besar beban di masing-masing fasa.

Perbedaan arus setiap fasa menyebabkan panas berlebih pada salah satu fasa, berkurangnya umur dan efisiensi pada transformator distribusi serta munculnya arus netral, sehingga menyebabkan *losses* akibat arus netral yang mengalir ke tanah. Untuk itu diperlukan suatu peralatan yang dapat memonitoring arus setiap fasa dan tegangan fasa tiap penghantar serta persentase ketidakseimbangan beban yang dilayani oleh transformator distribusi yang terhubung dengan *smartphone* atau PC/laptop operator melalui jaringan internet dengan tampilan berupa grafik,

sehingga dapat dipantau dimanapun dan kapanpun oleh operator serta dapat ditindak lanjuti dengan melakukan pemerataan beban.

Peralatan yang dapat memonitoring ketidakseimbangan beban transformator distribusi berbasis *internet of things* dalam penelitian ini berupa *prototype* yang diuji pada 3 unit transformator satu fasa yang dihubungkan menjadi transformator 3 fasa dengan belitan hubungan Wye dan menggunakan beberapa lampu pijar sebagai simulasi.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas maka penulis merumuskan permasalahan sebagai berikut:

- a. Bagaimana sistem perancangan *hardware prototype* transformator distribusi, sensor arus, sensor tegangan, Arduino Uno dan ESP8266-01 serta sistem perancangan *software* pada mikrokontroler Arduino Uno?
- b. Bagaimana pengaruh nilai arus setiap penghantar terhadap persentase ketidakseimbangan beban?
- c. Bagaimana kecepatan transfer data yang dibaca sensor arus dan sensor tegangan pada mikrokontroler Arduino Uno terhadap *Web server Thingspeak*?

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian ini antara lain:

- a. Merancang sebuah alat monitoring ketidakseimbangan beban pada *prototype* transformator distribusi berbasis *Internet of Things* menggunakan *Web Server Thingspeak*.

- b. Memberikan informasi pengukuran arus dan tegangan setiap penghantar pada sistem tiga fasa serta persentase ketidakseimbangan beban melalui internet.
- c. Mengimplementasikan konsep *Internet of Things* dengan menggunakan ESP8266-01 agar alat terkoneksi ke internet.

1.4. Batasan Masalah

Batasan-batasan masalah yang meliputi penelitian ini antara lain:

- a. Mengukur arus penghantar R, S, T dan tegangan fasa tiap penghantar serta menghitung persentase ketidakseimbangan beban.
- b. Menggunakan 3 unit transformator 1 fasa yang dihubungkan dengan belitan hubungan star sebagai prototipe transformator distribusi.
- c. Menggunakan beban resistif berupa lampu pijar sebagai simulasi.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dapat tercapai dalam penelitian ini adalah sebagai berikut:

- a. Mengaplikasikan Ilmu yang diperoleh selama menempuh pendidikan pada perkuliahan
- b. Mempermudah monitoring arus dan tegangan pada sistem 3 fasa serta persentase ketidakseimbangan beban dengan tampilan grafik melalui *smartphone* atau PC/laptop *user* menggunakan jaringan internet,
- c. Mengimplementasikan mata kuliah sistem distribusi bagi mahasiswa.
- d. Meningkatkan kecepatan dalam pengecekan nilai arus setiap penghantar dan tegangan fasa tiap penghantar serta persentase ketidakseimbangan beban.

BAB II

TINJAUAN PUSTAKA

2.1. Transformator Distribusi

Transformator merupakan suatu peralatan listrik yang dapat mentransfer dan mengubah energi listrik dari satu atau lebih rangkaian listrik ke rangkaian listrik yang lain melalui suatu gandengan magnet dan beroperasi berdasarkan prinsip induksi elektromagnetis dengan frekuensi yang sama dan perbandingan transformasi tertentu, dimana perbandingan tegangan antara sisi primer dan sisi sekunder berbanding lurus dengan perbandingan jumlah belitan dan berbanding terbalik dengan perbandingan arusnya.

Transformator distribusi berfungsi untuk menurunkan tegangan distribusi primer pada Jaringan Tegangan Menengah (JTM) menjadi tegangan distribusi sekunder pada Jaringan Tegangan Rendah (JTR). Transformator distribusi merupakan transformator *step down* 20 kV/ 0,4 kV, dengan tegangan fasa ke fasa 380 volt dan fasa ke netral 220 Volt.

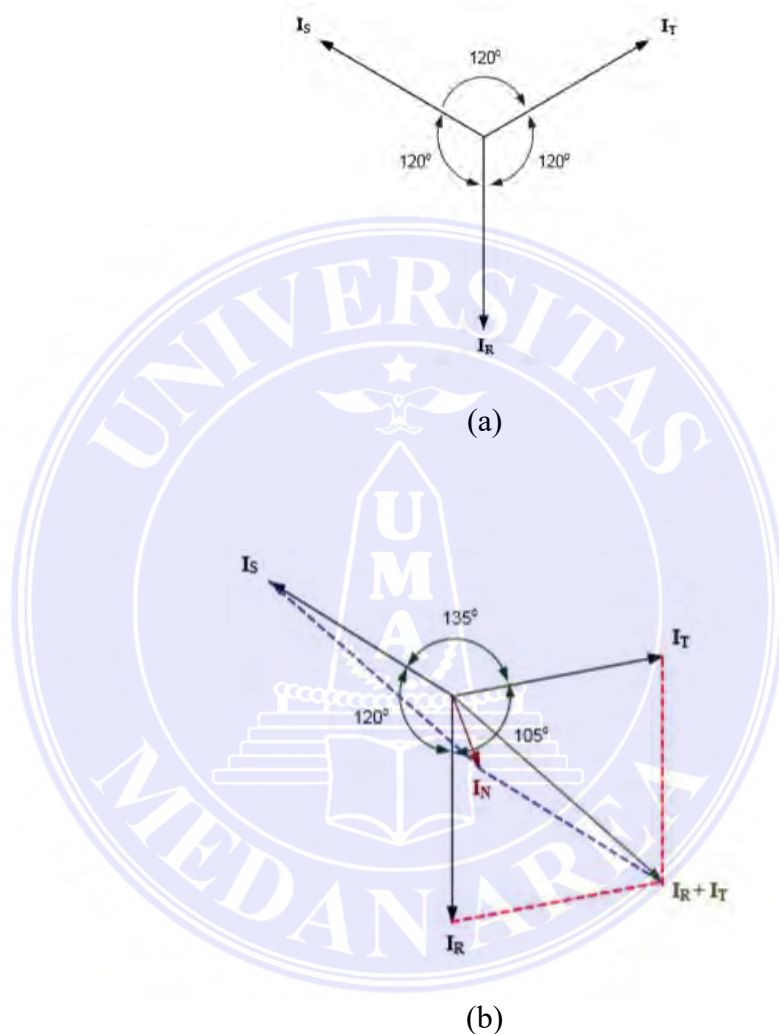
2.2. Ketidakseimbangan Beban

Keseimbangan beban dapat terjadi jika suatu keadaan dimana:

- a. Ketiga vektor arus/tegangan sama besar
- b. Ketiga vektor saling membentuk sudut 120° satu sama lain

Sedangkan ketidakseimbangan beban adalah keadaan dimana salah satu kedua syarat keadaan seimbang tidak terpenuhi. Kemungkinan keadaan tidak seimbang ada 3 yaitu:

- Ketiga vektor sama besar tetapi tidak membentuk sudut 120° satu sama lain.
- Ketiga vektor tidak sama besar tetapi membentuk sudut 120° satu sama lain.
- Ketiga vektor tidak sama besar dan tidak membentuk sudut 120° satu sama lain.



Gambar 2.1 (a) Vektor diagram arus seimbang, (b) Vector diagram arus tidak seimbang
(Sumber: Setiadji dkk. 2006)

Pada Gambar 2.1 (a) terlihat bahwa penjumlahan ketiga vektor arusnya (I_R , I_S , I_T) adalah sama dengan nol sehingga tidak muncul arus netral (I_N). Sedangkan pada Gambar 2.1 (b) menunjukkan vektor diagram arus yang tidak seimbang. Dapat dilihat bahwa penjumlahan ketiga vektor arusnya (I_R , I_S , I_T) tidak sama dengan nol

sehingga muncul sebuah besaran yaitu arus netral (I_N) yang besarnya bergantung dari beberapa besar faktor ketidakseimbangannya.

Untuk menghitung nilai ketidakseimbangan beban (dalam %) maka persamaan yang digunakan adalah sebagai berikut:

- a. Menentukan arus rata-rata beban pada transformator

$$I_{AVG} = \frac{I_R + I_S + I_T}{3} \quad (2.1)$$

Dimana:

I_{AVG} = Arus rata-rata (A)

I_R = Arus beban R (A)

I_S = Arus beban S (A)

I_T = Arus beban T (A)

- b. Menentukan koefisien masing-masing fasa

$$a = \frac{I_R}{I_{AVG}} \quad (2.2)$$

$$b = \frac{I_S}{I_{AVG}} \quad (2.3)$$

$$c = \frac{I_T}{I_{AVG}} \quad (2.4)$$

- c. Sehingga untuk persentase pembebanan Transformator (*Unbalance*)

$$U = \frac{|a - 1| + |b - 1| + |c - 1|}{3} \times 100\% \quad (2.5)$$

2.3. Internet of Things

Paradigma *Internet of things* (IoT) mengacu pada suatu jaringan yang terhubung satu sama dengan yang lain, seperti sensor atau *actuator* dan juga dilengkapi dengan *interface telecommunication*, dan unit pengolahan serta

penyimpanan. Komunikasi tersebut harus memungkinkan terintegrasi dengan tanpa hambatan. Dengan menggunakan internet, terjadi bentuk interaksi antara manusia dengan perangkat, yaitu komunikasi antara mesin ke mesin (M2M).

Peranan internet dalam kehidupan masyarakat modern sehari-hari menjadi bagian yang tak terpisahkan, sehingga kemudian muncul konsep IoT. IoT didefinisikan sebagai interkoneksi dari perangkat komputasi tertanam (Embedded computing device) yang teridentifikasi secara unik dalam keberadaan infrastruktur internet.

2.4. Sensor YHDC SCT 013-000

Sensor SCT 013-000 merupakan sensor yang berfungsi mengukur arus AC dengan jangkauan pembacaan arus AC 0-100 A dan hasil pembacaan 0-50 mA. Sensor ini cocok digunakan dengan *emonTx OpenEnergy*. Sensor SCT 013 000 umumnya digunakan untuk mendeteksi beban listrik, proteksi *Overload*. Dengan menggunakan sensor ini sebagai sensor arus, beban tidak perlu dihubungkan secara seri dengan sensor namun hanya dengan menjepitkan inti besi yang terdapat pada sensor dengan kawat penghantar yang akan diukur arusnya.



Gambar 2.2. Sensor arus SCT 013-000

(Sumber: https://yhdctransformer.en.ec21.com/YHDC_SCT_013_000_100A--9226490_9254491.html)

Dalam proses induksi, arus yang melewati kawat sisi primer menciptakan medan magnet pada inti *ferrite* dari CT sensor. Hal ini dapat dilihat seperti pada Gambar 2.2. Arus listrik kecil yang proporsional dihasilkan dari kawat pada sisi sekunder yang mengelilingi inti tersebut. Didalam sensor ini terdapat dioda zener yang berfungsi membatasi keluaran tegangan jika terjadi pemutusan mendadak dari beban.

2.5. Sensor Tegangan ZMPT 101B

Sensor tegangan ZMPT 101B (Gambar 2.3) dirancang menggunakan transformator sehingga hanya bisa digunakan membaca tegangan AC. Sensor ini dilengkapi dengan rangkaian *summing amplifier* dan tidak memerlukan sampling saat proses pembacaan sensor karena jika ADC belum bisa membaca sinyal tegangan negatif dan tanpa menaikkan *offset*-nya ke 2,5 volt, sensor ini dapat menaikkan sendiri tanpa harus memasang rangkaian eksternal *summing amplifier*.



Gambar 2.3. Sensor tegangan ZMPT 101B

(Sumber: <https://mybookshelvesweb.files.wordpress.com/2017/09/zmpt101b-module-1-350x350.jpg?w=263&h=263>)

Transformator AC ZMPT 101B memiliki 4 pin, pin 1 dan pin 2 berfungsi sebagai input tegangan, pin 3 dan pin 4 berfungsi sebagai output tegangan yang diturunkan. ZMPT 101B memiliki output tegangan ADC 512 hingga 1024 yang

linier dari 0 V sampai 1000 V. Namun pengaturan maksimal pada trimpot yang ada pada modul sensor ZMPT 101B hanya mengeluarkan ADC 640 yang mengacu pada batas efektif tegangan yang bisa dibaca adalah 250V AC.

2.6. Thingspeak

Thingspeak (Gambar 2.4) merupakan *platform* IoT yang dapat mengembangkan aplikasi IoT dan dapat mengumpulkan serta menyimpan data pada cloud. Output yang dihasilkan dalam bentuk grafik ditingkat web. *Thingspeak* memiliki hubungan dekat dengan MathWorks, Inc dan dokumentasi *Thingspeak* dimasukkan ke situs dokumentasi MathWorks. Sehingga juga menyediakan beberapa *toolbox* MATLAB dan menggunakan akun MATLAB yang telah berlisensi.



Gambar 2.4. *Thingspeak*

(Sumber: <https://thingspeak.com>)

Untuk dapat menggunakan *platform thingspeak*, pengguna terlebih dahulu membuat akun dan menentukan *channel* pada akun tersebut. Selanjutnya *platform thingspeak* akan memberikan API (*Application Programming Interface*) key yang

diatur di dalam pemrograman mikrokontroler, sehingga bisa mengirimkan data dari sensor ke *Thingspeak*.

2.7. Arduino UNO

Arduino UNO (Gambar 2.5) merupakan sebuah *board* mikrokontroller berbasis Atmega328. Arduino Uno memiliki 14 pin input/output, 6 diantaranya digunakan sebagai output PWM, 6 input analog, osilator kristal 16 MHz, konektor USB, colokan listrik, *header* ICSP dan tombol reset. Untuk penggunaan Arduino dapat memulai dengan menghubungkannya ke komputer dengan kabel USB atau dapat menggunakan adaptor AC-DC dan baterai.



Gambar 2.5 Arduino Uno

(Sumber: Harry Yuliansyah. 2016)

Arduino adalah sebuah *platform* dari komputasi fisik yang bersifat *open source*. Arduino lebih dari sekedar alat pengembangan dan merupakan kombinasi dari perangkat keras canggih, bahasa pemrograman, dan *Integrated Development Environment* (IDE). IDE adalah perangkat lunak yang berguna untuk menulis, meng-*compile* menjadi kode biner, dan mengunggahnya ke memori mikrokontroler.

2.8. ESP8266-01

Modul ESP8266 (Gambar 2.6) merupakan modul Wi-Fi yang mendukung penuh penggunaan TCP/IP. Modul ini diproduksi oleh *Espressif Chinese Manufacturer*. AI-Thinker Manufaktur sebagai pihak ketiga dari modul ini merilis modul ESP8266-01 pada tahun 2014 dengan harga yang murah, penggunaan daya yang rendah dan dimensi modul yang kecil. Dalam konfigurasinya, modul ini menggunakan *AT-Command*. *AT Command* ini dapat berfungsi apabila ESP8266 sudah terinstall ESP8266 *AT Firmware* atau *firmware* lain yang mendukung *AT firmware*.



Gambar 2.6 Modul Wi-Fi ESP8266-01

(Sumber: Harry Yuliansyah, 2016)

Espressif mengeluarkan *software development kit* (SDK) yang memungkinkan lebih banyak *developer* untuk mengembangkan modul ini. Modul ESP8266-01 memiliki *form factor* 2x4 DIL yang dengan dimensi 14,3 x 24,8 mm. Catu daya yang dibutuhkan adalah 3,3 Volt.

2.9. Manajemen *Bandwidth*

Bandwidth adalah ukuran jumlah data yang dapat dikirim oleh koneksi melalui jaringan. *Bandwidth* menunjukkan total jarak atau rentang antara sinyal tinggi dan rendah dari saluran komunikasi. *Bandwidth* digunakan untuk transmisi data dalam

jaringan komputer. Artinya, jumlah data yang ditransfer dari satu titik pada waktu tertentu ke yang lain. Jenis *bandwidth* ini biasanya diukur dalam bps (bit per detik), bps (byte/detik).

Manajemen *bandwidth* untuk meningkatkan kualitas layanan (QoS) menyatakan bahwa *bandwidth* adalah "jumlah data yang dapat dikirim melalui koneksi jaringan, kapasitas saluran informasi, atau jumlah yang menunjukkan kapasitas maksimum perangkat dapat mengirim informasi dalam detik". Manajemen *bandwidth* adalah alat untuk mengelola dan mengoptimalkan berbagai jenis jaringan dengan menerapkan layanan QoS yang membantu menentukan jenis lalu lintas jaringan. Definisi QoS adalah suatu fungsi yang menggambarkan tingkat performansi komunikasi data. Alokasi *bandwidth* yang tepat adalah cara untuk memberikan jaminan kualitas untuk layanan jaringan QoS. Manajemen *bandwidth* memberikan kemampuan untuk mengelola *bandwidth* jaringan dan memberikan tingkat layanan sesuai dengan permintaan dan prioritas.

BAB III

METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

3.1.1. Waktu Penelitian

Waktu yang diperlukan dalam pengerjaan penelitian ini kurang lebih tiga bulan. Hal ini dapat ditunjukkan seperti tampak pada Tabel 3.1. berikut.

Tabel 3.1. Jadwal pelaksanaan kegiatan

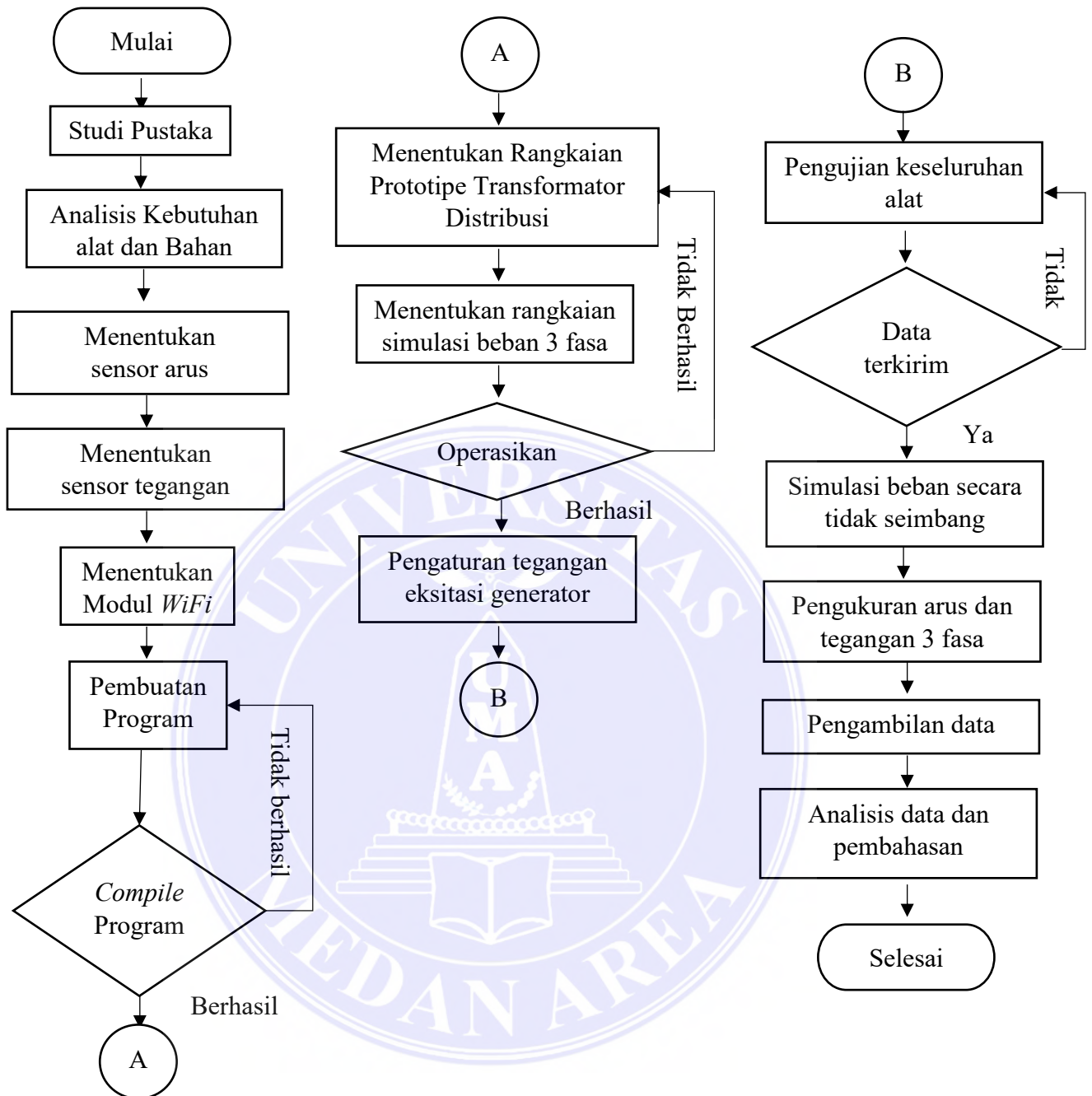
No.	Kegiatan Penelitian	Bulan Ke											
		I				II				III			
		1	2	3	4	1	2	3	4	1	2	3	4
1.	Analisis Kebutuhan dan Perancangan												
2.	Pengumpulan Alat dan Bahan												
3.	Pembuatan Alat												
4.	Pengujian Alat												
5.	Pengumpulan Data												
6.	Analisa Data												
7.	Penulisan Laporan												

3.1.2. Tempat Penelitian

Adapun penelitian ini telah dilakukan di Laboratorium Konversi Energi Listrik Program Studi Teknik Elektro, Fakultas Teknik Universitas Medan Area.

3.2. Prosedur Kerja

Penelitian rancang bangun prototipe monitoring ketidakseimbangan beban pada transformator distribusi berbasis *internet of things* ini meliputi beberapa tahapan yakni seperti yang ditunjukkan diagram alir pada Gambar 3.1 berikut.



Gambar 3.1 Diagram alir penelitian

3.2.1. Analisis Kebutuhan Alat dan Bahan

Kebutuhan perangkat yang penulis gunakan di dalam penelitian ini terbagi menjadi dua bagian, yakni memanfaatkan perangkat yang ada di Laboratorium Konversi Energi Listrik Program Studi Teknik Elektro UMA dan perangkat dari penulis gunakan sendiri dan dapat dilihat pada Tabel 3.2 dan Tabel 3.3:

Tabel 3.2. Komponen di laboratorium yang digunakan

No	Komponen	Spesifikasi	Satuan/Luas
1	Transformator	1 fasa	3 unit
2	Multimeter digital	-	6 unit
3	Beban resistif	100 watt	15 unit
4	Autotransformator	1 fasa	1 unit
5	Motor kopel generator	-	1 unit, 1 unit
7	Kabel Jumper	-	secukupnya

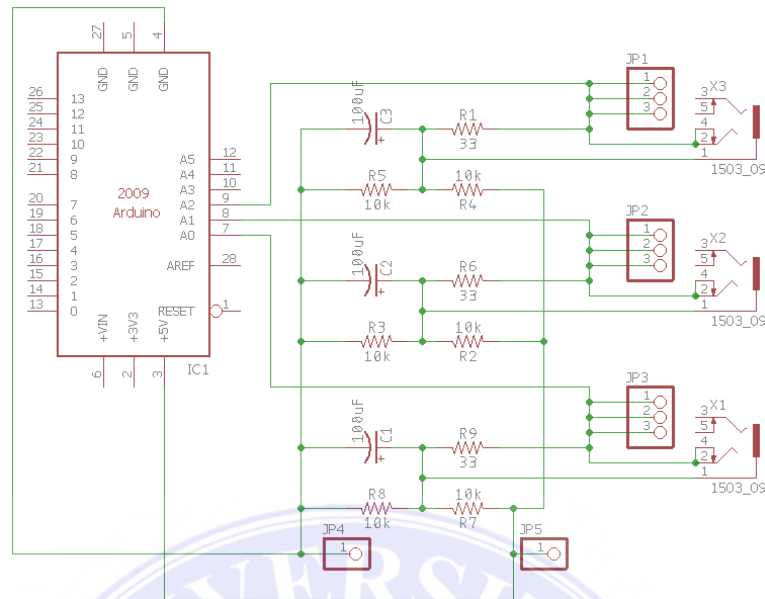
Tabel 3.3. Komponen yang penulis gunakan

No	Komponen	Spesifikasi	Satuan/ Luas
1	Mikrokontroler	Arduino Uno	1 unit
2	Sensor arus	SCT 013-000	3 unit
3	Sensor Tegangan	ZMPT 101B	3 unit
4	Modul <i>WiFi</i>	ESP8266-01	1 unit
5	Resistor	33 Ohm	3 unit
		10k Ohm	6 unit
6	Elektrolit Kapasitor	100 μ f/16 V	3 unit
7	Socket Female Stereo Jack	3,5 mm	3 unit
8	Laptop	Hewlett Packard	1 unit
9	<i>Smartphone</i>	Oppo A5S	1 unit
10	PCB bolong	IC	18 cm x 7 cm
		Titik-titik	16,4 cm x 8,4 cm
11	Kabel Jumper	Male to male	Secukupnya
		Female to male	
12	Pin header	Male 2,54 mm	29 pin
		Female 2,54 mm	8 pin
13	Modul FTDI RS232	-	1 unit

3.2.2. Menentukan Sensor Arus

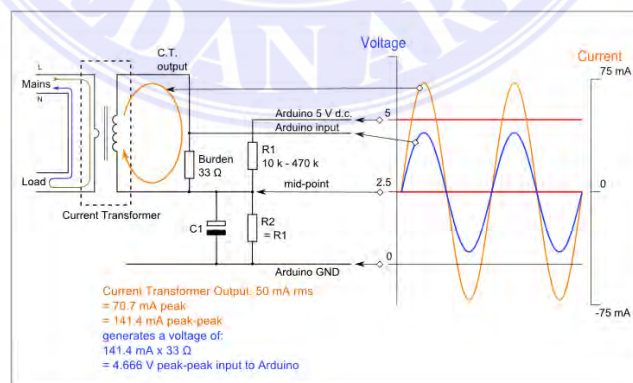
Sensor arus yang digunakan untuk mengukur nilai arus beban pada fasa R, fasa S dan fasa T adalah sensor arus SCT 013-000 dengan maksimum pengukuran arus AC yakni 100A.

Adapun gambar rangkaian sensor arus yang digunakan dalam pengujian prototipe dapat dilihat seperti tampak pada Gambar 3.2 berikut.



Gambar 3.2. Rangkaian sensor arus menggunakan sensor arus SCT 013-000 pada fasa R, fasa S dan fasa T
(Sumber: Perancangan menggunakan Eagle)

Berdasarkan Gambar 3.2 di atas, dapat dijelaskan bahwa dalam menggunakan setiap sensor arus dibutuhkan rangkaian pengkondisi sinyal dimana X1, X2 dan X3 merupakan Jack Connector 3,5 mm untuk menghubungkan sensor ke rangkaian pengkondisi sinyal yang terhubung dengan Pin Analog A0, A1 dan A2 pada Arduino Uno.



Gambar 3.3. Skema pengkondisi sinyal sensor arus
(Sumber: <https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/interface-with-arduino?redirected=true>)

Kapasitor elektrolit digunakan untuk mengkompensasi perubahan mendadak pada gelombang atau lonjakan tegangan dari sensor. R1 dan R2 merupakan pembagi tegangan agar dapat bekerja pada EmonTx. Sedangkan resistor dengan nilai 33 ohm merupakan resistor beban berdasarkan perhitungan sebagai berikut :

$$\text{Arus Puncak Primer} = \text{Arus RMS} \times \sqrt{2} = 100 \text{ A} \times 1,414 \text{ A}$$

$$\text{Arus puncak sekunder} = \frac{\text{ arus puncak primer}}{\text{Jumlah Putaran}} = \frac{1,414 \text{ A}}{2000} = 0,0707 \text{ A}$$

Untuk memaksimalkan resolusi pengukuran, tegangan yang melalui resistor beban pada arus puncak harus sama dengan setengah dari tegangan referensi analog arduino(AREF/2) yakni 5 V maka:

$$\frac{AREF}{2} = \frac{5}{2} = 2,5 \text{ Volt}$$

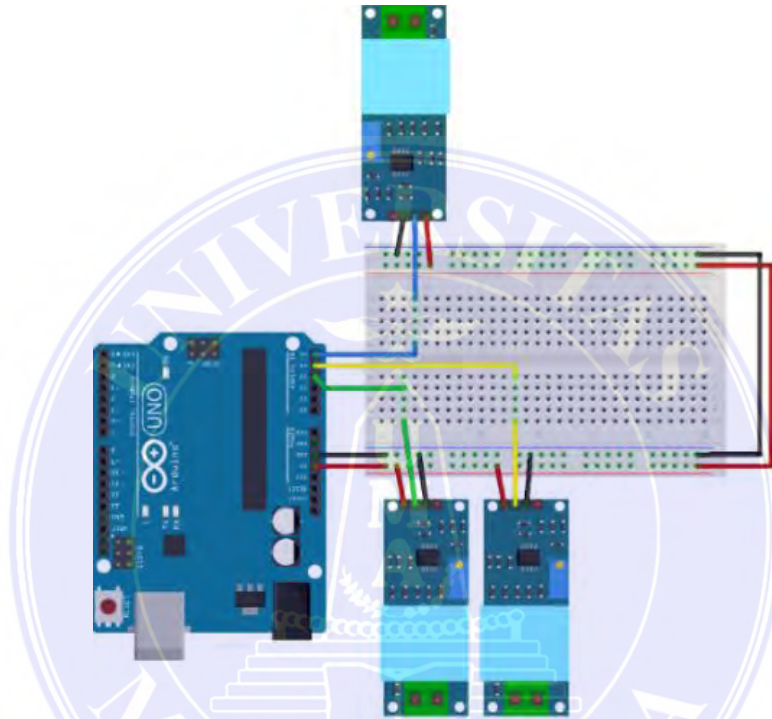
$$\text{Resistor Beban} = \frac{AREF/2}{\text{Arus puncak Sekunder}} = \frac{2,5 \text{ V}}{0,0707 \text{ A}} = 35,4 \Omega$$

Nilai 35,4 Ohm bukan merupakan nilai resistor pada umumnya, maka digunakan resistor dengan nilai terdekat yakni 33 ohm. Hal ini juga berlaku pada rangkaian pengkondisi sinyal untuk kedua sensor arus lainnya pada Jack Connector X2 dan X3.

Cara kerja sensor ini berfungsi sebagai koil induksi yang mendeteksi perubahan medan magnet yang terjadi di sekeliling konduktor pembawa arus. Dengan mengukur jumlah arus yang dibangkitkan oleh koil sehingga dapat menghitung jumlah arus yang melewati konduktor tersebut (prinsip medan magnet pada transformator).

3.2.3. Menentukan Sensor Tegangan

Sensor tegangan yang digunakan untuk mengukur tegangan *line to neutral* pada transformator 3 fasa hubungan Wye, yakni R-N, S-N dan T-N adalah sensor tegangan ZMPT 101B. Berikut ini merupakan gambar rangkaian sensor tegangan ZMPT 101B yang digunakan dalam pengujian *prototype*.



Gambar 3.4. Rangkaian sensor tegangan fasa R, fasa S dan fasa T

Keterangan Gambar 3.4:

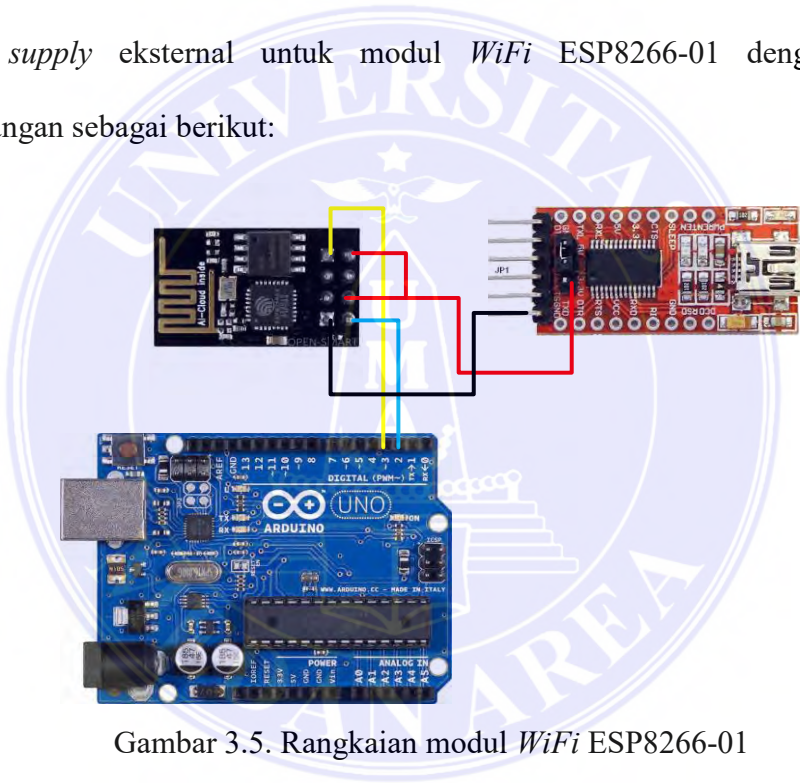
- a. Pin 5V arduino terhubung ke setiap pin VCC ZMPT 101B (kabel merah),
- b. Pin GND arduino terhubung ke GND ZMPT 101B (kabel hitam),
- c. Pin A3 arduino terhubung ke pin OUT ZMPT 101B fasa R-N (kabel hijau),
- d. Pin A4 arduino terhubung ke pin OUT ZMPT 101B fasa S-N (kabel kuning),
- e. Pin A5 arduino terhubung ke pin OUT ZMPT 101B fasa T-N (kabel biru).

Prinsip kerja dari sensor ini adalah dengan menurunkan tegangan masukan menggunakan transformator *step down* dan rangkaian op-amp sebagai penguat,

sehingga dapat mengubah tegangan masukan yang besar menjadi rendah, stabil dan aman dihubungkan ke arduino. Tegangan keluaran dapat dikalibrasi dengan menggunakan trimpot yang terpasang pada modul.

3.2.4. Menentukan Modul WiFi

Untuk mengirimkan data pembacaan sensor tegangan dan sensor arus dari Arduino Uno ke *Thingspeak*, penulis menggunakan modul FTDI RS232 sebagai *power supply* eksternal untuk modul *WiFi* ESP8266-01 dengan gambar perancangan sebagai berikut:



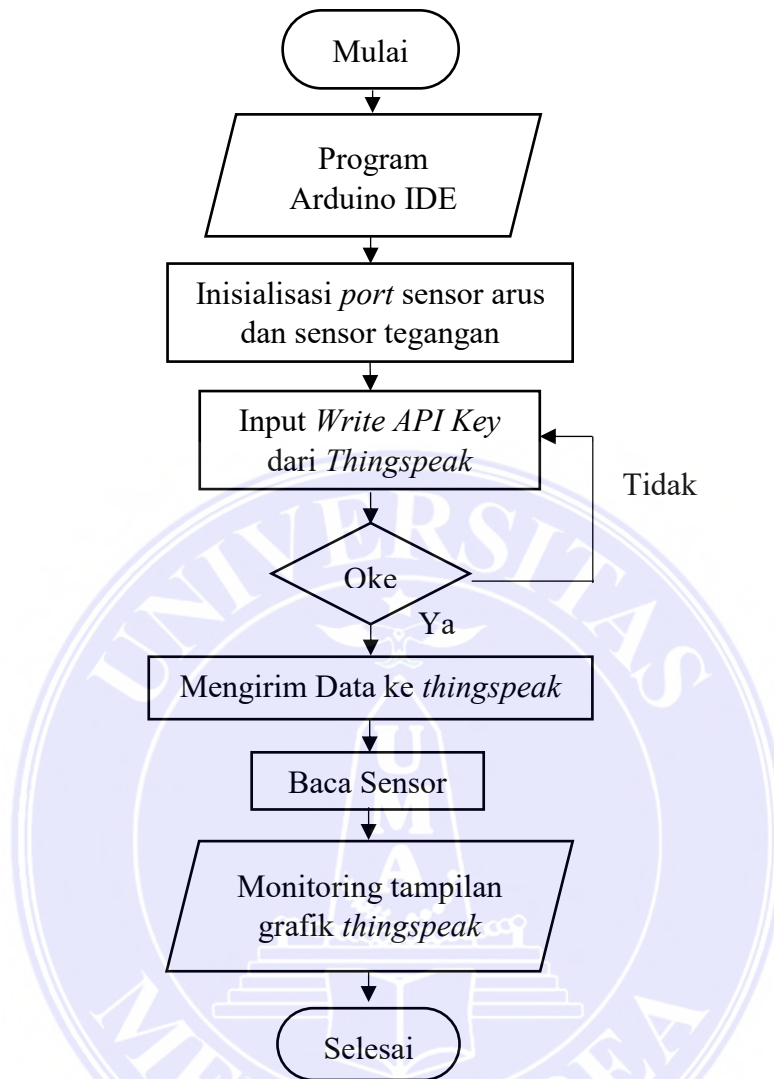
Gambar 3.5. Rangkaian modul *WiFi* ESP8266-01

Keterangan Gambar 3.5 ditunjukkan dalam Tabel 3.4 berikut ini.

Tabel 3.4. Konfigurasi ESP8266-01 ke Arduino Uno dan FTDI RS232

ESP8266-01 ke FTDI RS232		ESP8266-01 ke Arduino UNO	
VCC ; CH_PD	3V3	RX	PIN 3
Ground	Ground	TX	PIN 2

3.2.5. Pembuatan Program



Gambar 3.6. Flowchart perancangan perangkat lunak

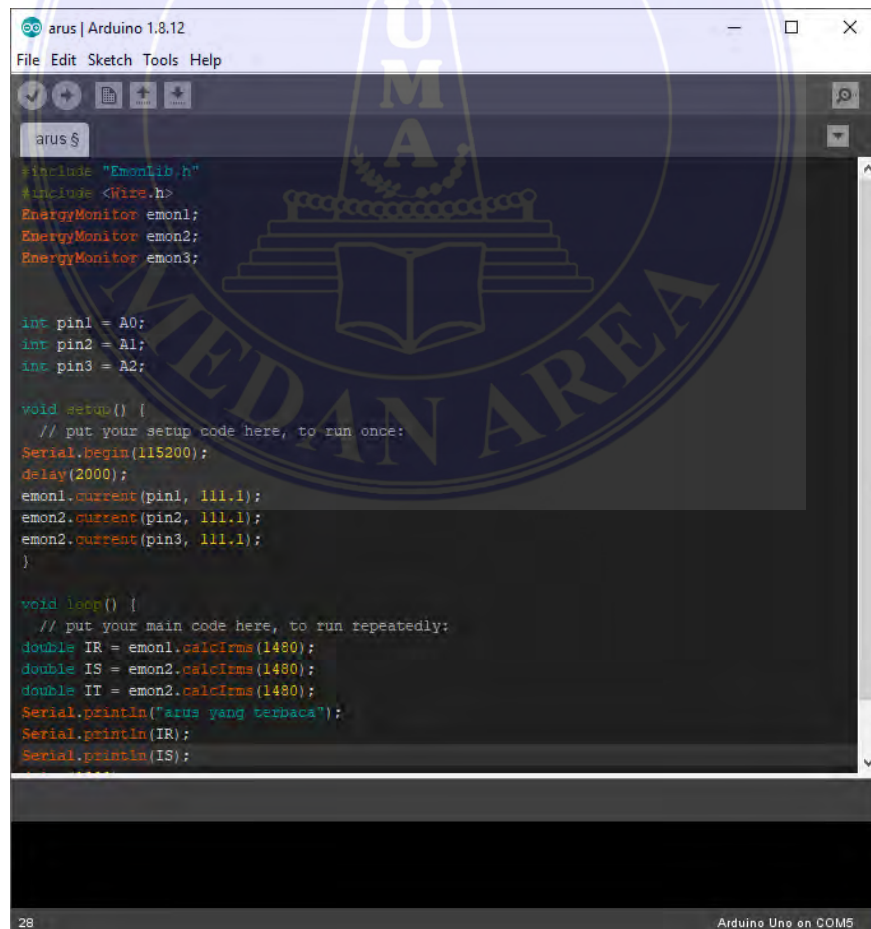
Berdasarkan diagram alir pada Gambar 3.6, proses dalam pembuatan program dimulai dengan membuka *software* arduino IDE dan menginisialisasi *port* sensor arus, sensor tegangan dan proses internal ESP8266-01. Kemudian mengakuratkan pembacaan nilai sensor arus dan sensor tegangan serta membuat algoritma perhitungan ketidakseimbangan beban dengan memasukkan persamaan (2-1) hingga persamaan (2-5). Selanjutnya *Sign Up* untuk membuat akun *Thingspeak*. Kemudian membuat *channel* baru serta nama dan jumlah grafik. Selanjutnya menyalin *Write API (Application Programming Interface) key* ke pemrograman

pada *software* Arduino IDE. Data arus tiap fasa dan tegangan fasa tiap penghantar serta persentase ketidakseimbangan beban yang dibaca sensor arus dan sensor tegangan akan diolah oleh Arduino Uno dan akan dikirimkan oleh ESP8266-01 ke *web server Thingspeak* yang sudah terhubung jaringan internet dalam bentuk tampilan grafik secara *online*.

3.2.5.1. Menentukan Inisialisasi *Port* Sensor Arus dan Sensor Tegangan

Untuk menentukan inisialisasi *port* sensor arus dan sensor tegangan pada Arduino Uno, berikut ini merupakan pemrograman inisialisasi port sensor arus dan sensor tegangan didalam *software* Arduino IDE yang ditunjukkan pada Gambar 3.7.

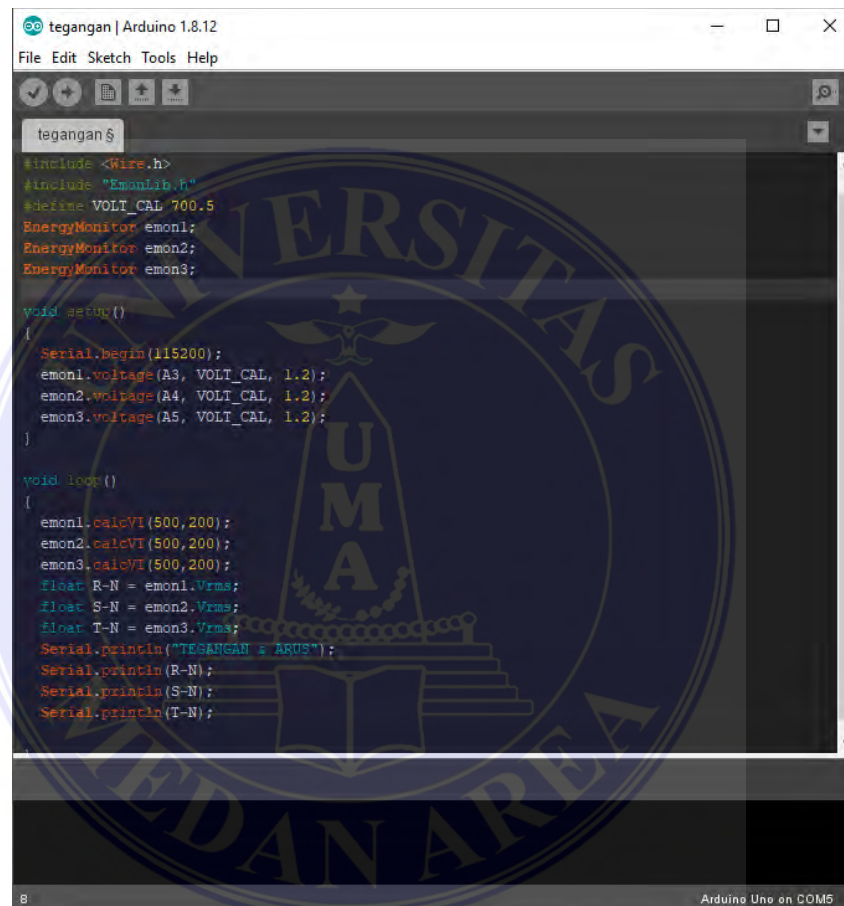
a. Menentukan inisialisasi *Port* sensor arus



Gambar 3.7. Inisialisasi *port* sensor arus pada *software* Arduino IDE

Dari Gambar 3.7, ditunjukkan bahwa pin A0 pada Arduino Uno diinisialisasikan sebagai pin1, pin A1 Arduino Uno diinisialisasikan sebagai pin2 dan pin A2 Arduino Uno diinisialisasikan sebagai pin3. Pin A0, A1 dan A2 arduino uno digunakan sebagai pembaca pengukuran arus fasa R, fasa S dan fasa T.

b. Menentukan inisialisasi *port* sensor tegangan



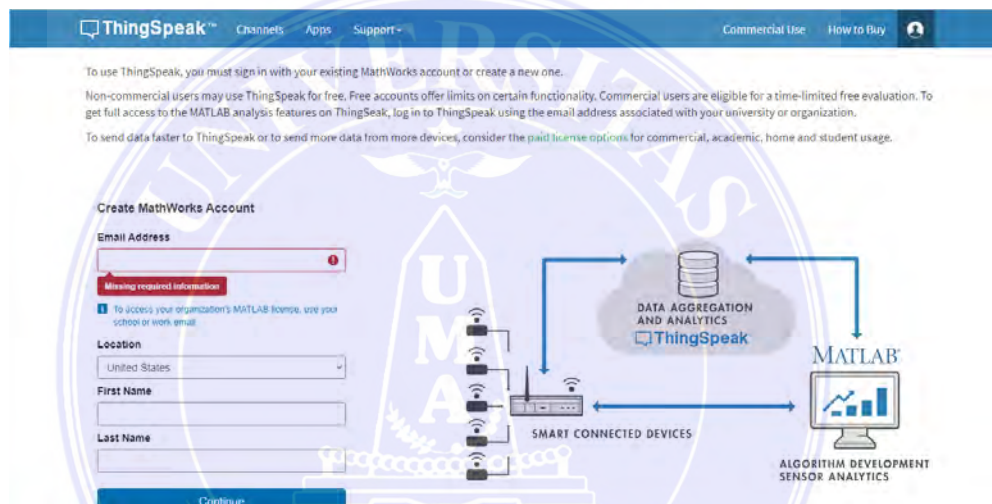
Gambar 3.8. Inisialisasi *port* sensor tegangan pada *software* Arduino IDE

Dari Gambar 3.8, ditunjukkan bahwa pin A3 arduino diinisialisasikan sebagai pengukur tegangan *line to neutral* R-N, pin A4 arduino diinisialisasikan sebagai pengukur tegangan *line to neutral* S-N. Pin A5 arduino diinisialisasikan sebagai pengukur tegangan *line to neutral* T-N.

3.2.5.2. Menentukan *Channel Thingspeak*

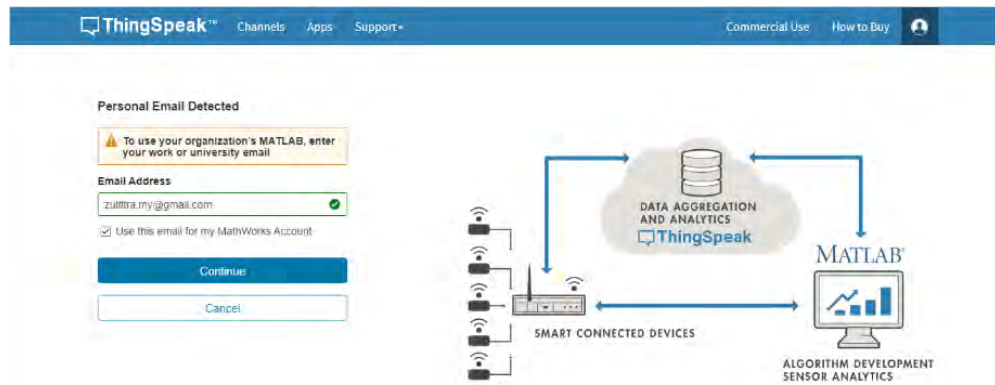
Untuk membuat *channel Thingspeak* yang akan digunakan sebagai media monitoring pengiriman data oleh sensor arus dan sensor tegangan pada tiap fasa, terlebih dahulu dilakukan beberapa tahapan sebagai berikut:

- a. Membuat akun *Thingspeak* pada link <https://thingspeak.com/login> dengan memasukkan data pengguna berupa email, lokasi, nama depan dan nama belakang.



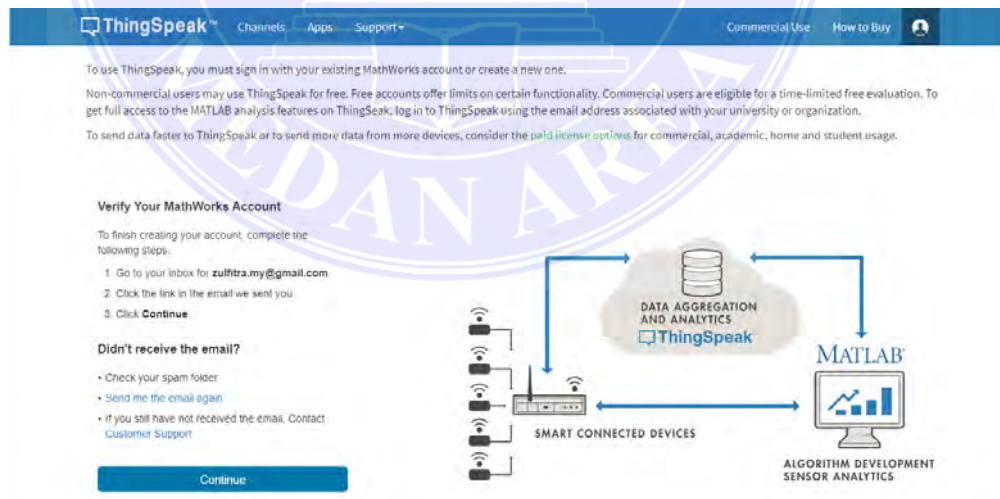
Gambar 3.9. Halaman login *thingspeak*

- b. Setelah pengisian data pengguna selesai, pengguna diarahkan ke halaman selanjutnya untuk memastikan alamat email yang didaftarkan agar dapat diverifikasi



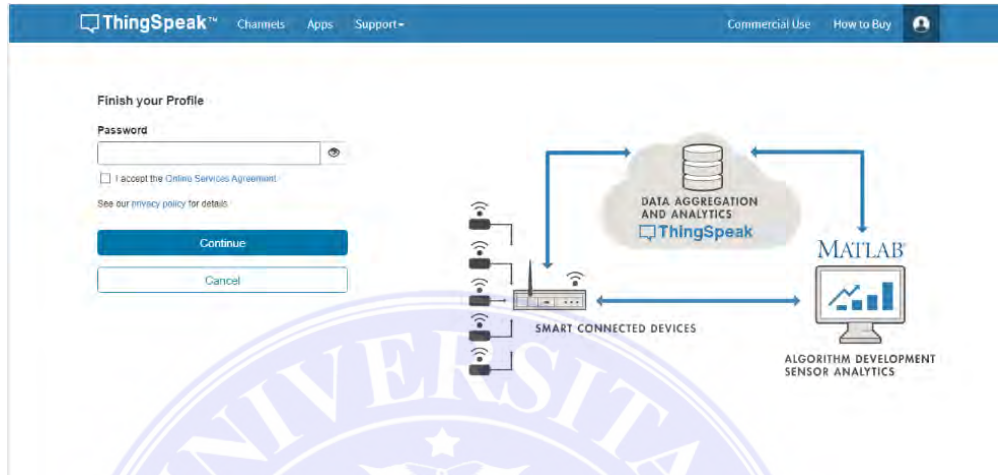
Gambar 3.10. Halaman verifikasi email

- c. Kemudian pada halaman log in selanjutnya email pengguna yang didaftarkan akan menerima pesan masuk yang dikirimkan oleh service@account.mathworks.com setelah mengklik continue seperti pada Gambar 3.11 dibawah ini. Klik link pada pesan masuk yang dikirimkan agar dapat diverifikasi.



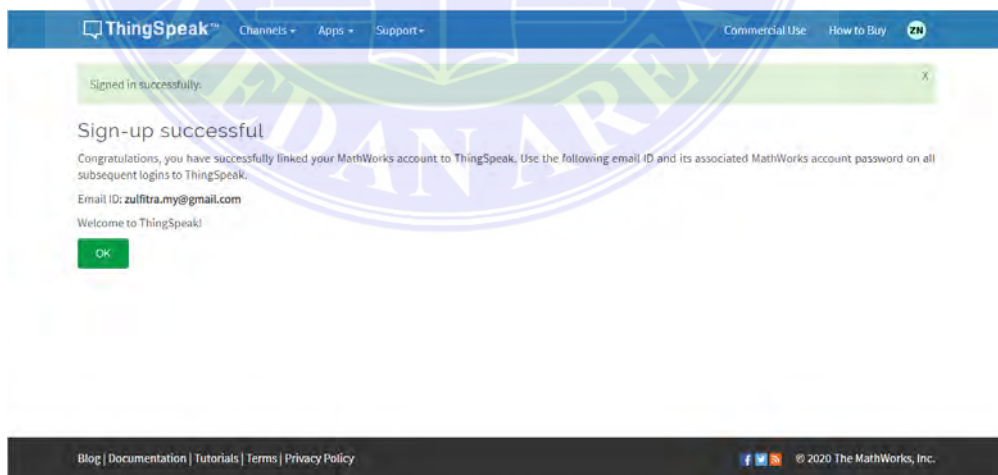
Gambar 3.11. Verifikasi akun MathWorks

- d. Setelah pengguna mengklik link pada pesan email pengguna, maka akan muncul halaman untuk menyelesaikan profil pengguna dengan mengisi password pada akun *thingspeak* yang akan digunakan.



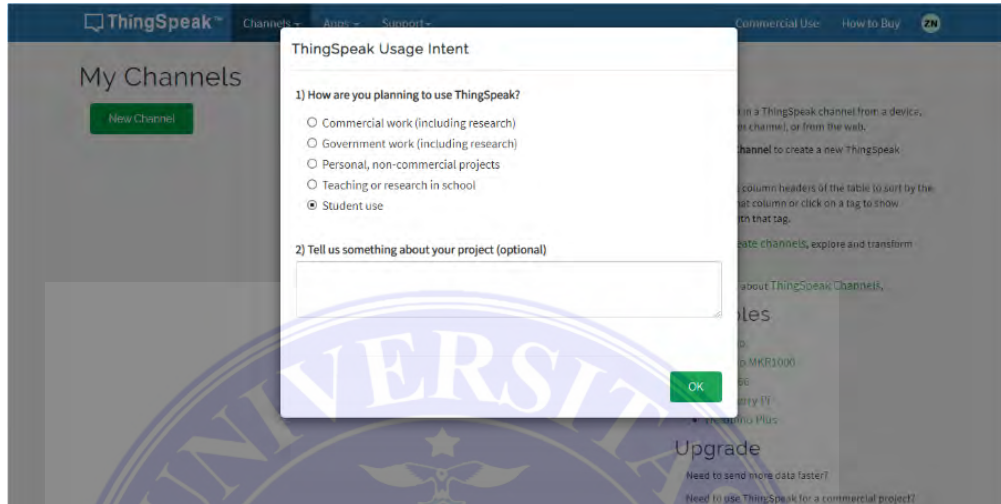
Gambar 3.12. Halaman pembuatan password akun *thingspeak*

- e. Pada halaman selanjutnya, jika password untuk akun *thingspeak* pengguna telah dinyatakan aman maka akun *thingspeak* berhasil dibuat dengan email yang tertera seperti tampak pada Gambar 3.13 berikut.



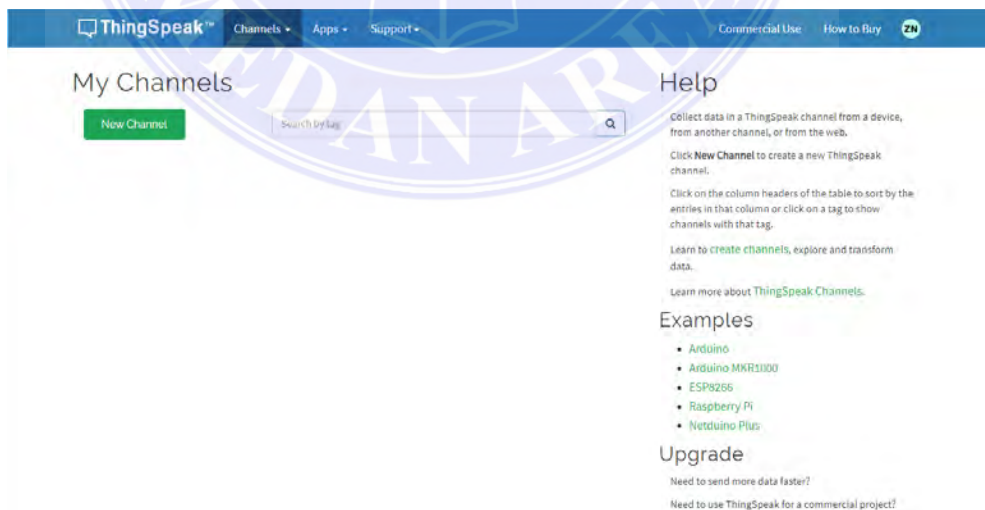
Gambar 3.13. Halaman pembuatan akun *thingspeak* sukses

- f. Pada halaman selanjutnya, pengguna akan diberi opsi status pengguna dalam menggunakan akun *thingspeak* digunakan, kemudian klik OK setelah opsi dipilih.



Gambar 3.14. Opsi status pengguna akun *thingspeak*

- g. Untuk membuat *channel* baru, klik pada tab *Channels*, kemudian pilih *My Channel*, maka *thingspeak* akan menampilkan *channel* yang sudah pernah dibuat pengguna. Jika *channel* belum dibuat maka klik *New Channel*.

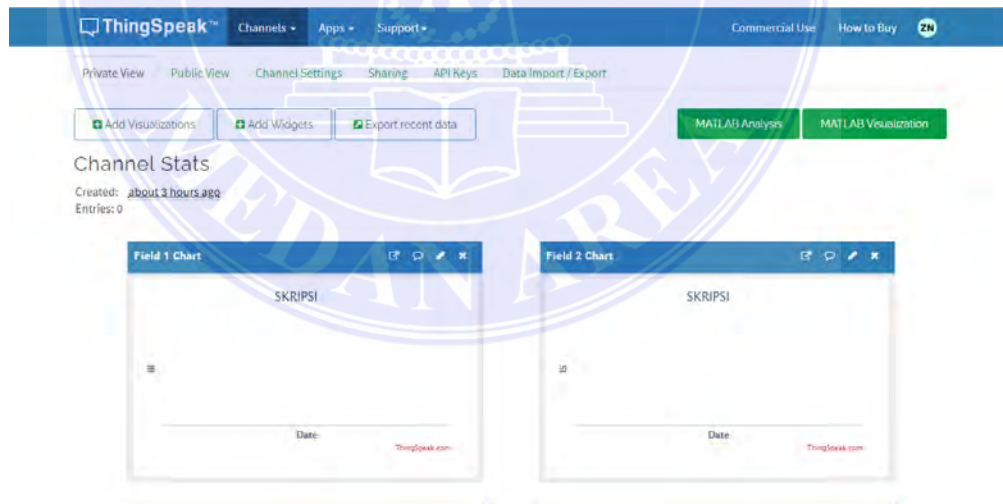


Gambar 3.15. Halaman pembuatan *channel* baru

- h. Pada halaman *New Channel*, isikan identitas pada *channel* yang akan digunakan seperti tampak pada Gambar 3.16 berikut.

Gambar 3.16. Halaman *form* identitas *channel* baru.

- i. Setelah pengguna mengisi data identitas pada *channel* baru, pengguna dapat melihat tampilan *channel* baru dalam bentuk grafik.

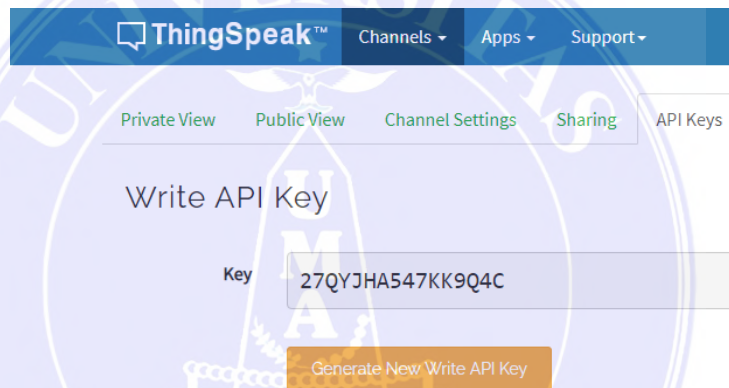


Gambar 3.17. Tampilan status *channel*

- j. Pengguna juga dapat melihat *Channel ID* yang digunakan dalam menghubungkan *thingspeak* dengan perangkat pengguna.

Gambar 3.18. *Channel ID*

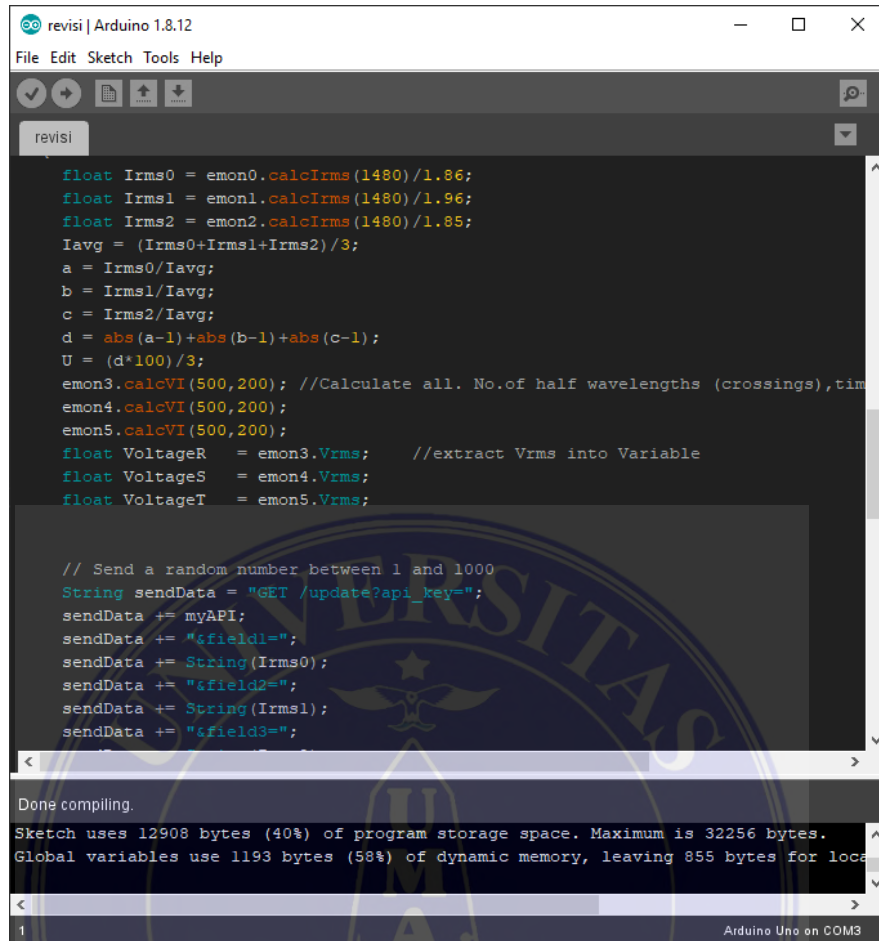
- k. Untuk menghubungkan *prototype* monitoring ketidakseimbangan beban dengan *channel* pada akun *thingspeak* yang digunakan, maka pengguna menyalin *Write API Key* pada *channel* ke dalam pemrograman *Software* Arduino IDE.

Gambar 3.19. *Write API Key channel prototype monitoring*

3.2.6. Compile Program

Berikut ini merupakan langkah-langkah dalam *compile* program yang telah dibuat:

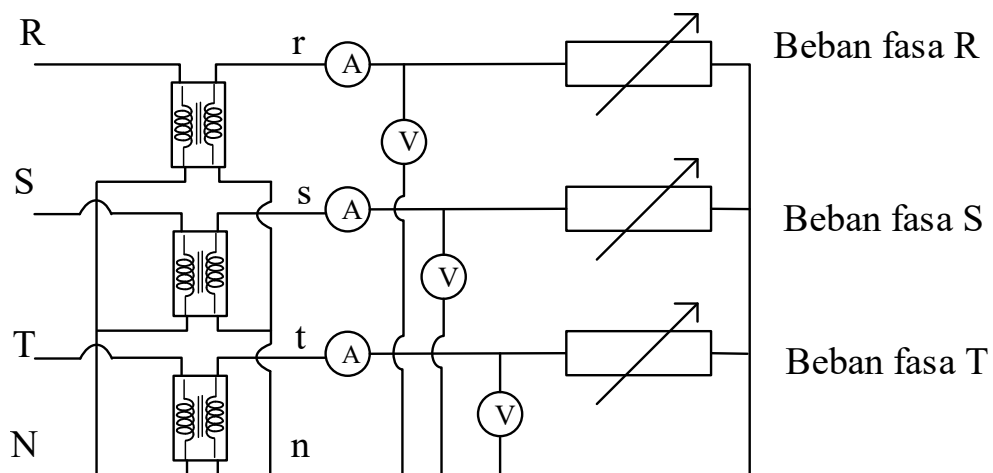
- Membuka *software* Arduino IDE di Laptop
- Klik *Sketch* → *Verify*, maka akan muncul kotak dialog untuk menyimpan *file project*,
- Klik *Save*, *compile* program diproses hingga muncul status *Done compiling* jika tidak ada *error*. Dapat dilihat pada Gambar 3.20 berikut.



Gambar 3.20. Compile program

3.2.7. Menentukan Rangkaian Prototipe Transformator Distribusi

Untuk melakukan pengujian ketidakseimbangan beban pada *prototype* monitoring ketidakseimbangan beban pada transformator distribusi, maka diperlukan 3 unit transformator 1 fasa yang dihubungkan secara wye 4 penghantar pada sisi primer dan sisi sekunder yang terhubung dengan simulasi beban (Gambar 3.21) berupa beberapa lampu pijar yang bersifat resistif di Laboratorium Konversi Energi Listrik Program Studi Teknik Elektro Fakultas Teknik Universitas Medan Area.



Gambar 3.21. Rangkaian 3 fasa hubungan wye dengan beban

Multimeter digital dengan *selector* pembacaan arus, dipasang secara seri dengan kabel setiap fasa. Sedangkan sensor tegangan dan multimeter dengan *selector* pembaca tegangan dihubungkan secara paralel seperti tampak pada Gambar 3.21. Dengan menggunakan SCT 013-000 sebagai sensor arus dalam menguji *prototype*, tidak perlu dihubungkan secara seri dengan beban namun cukup dengan menjepitkan inti besi yang terdapat pada sensor dengan kabel penghantar pada fasa R, S dan T.

Penggunaan alat ukur multimeter digital digunakan untuk mengukur arus dan tegangan sebagai perbandingan nilai pembacaan dengan sensor arus dan sensor tegangan pada *prototype*.

3.2.8. Menentukan Rangkaian Simulasi Beban 3 Fasa

Simulasi Beban yang digunakan merupakan beban resistif sebanyak 5 lampu pijar pada setiap jalur fasa yang digunakan dan merupakan perangkat yang ada di Laboratorium Konversi Energi Listrik, Program Studi Teknik Elektro Fakultas Teknik Universitas Medan Area.



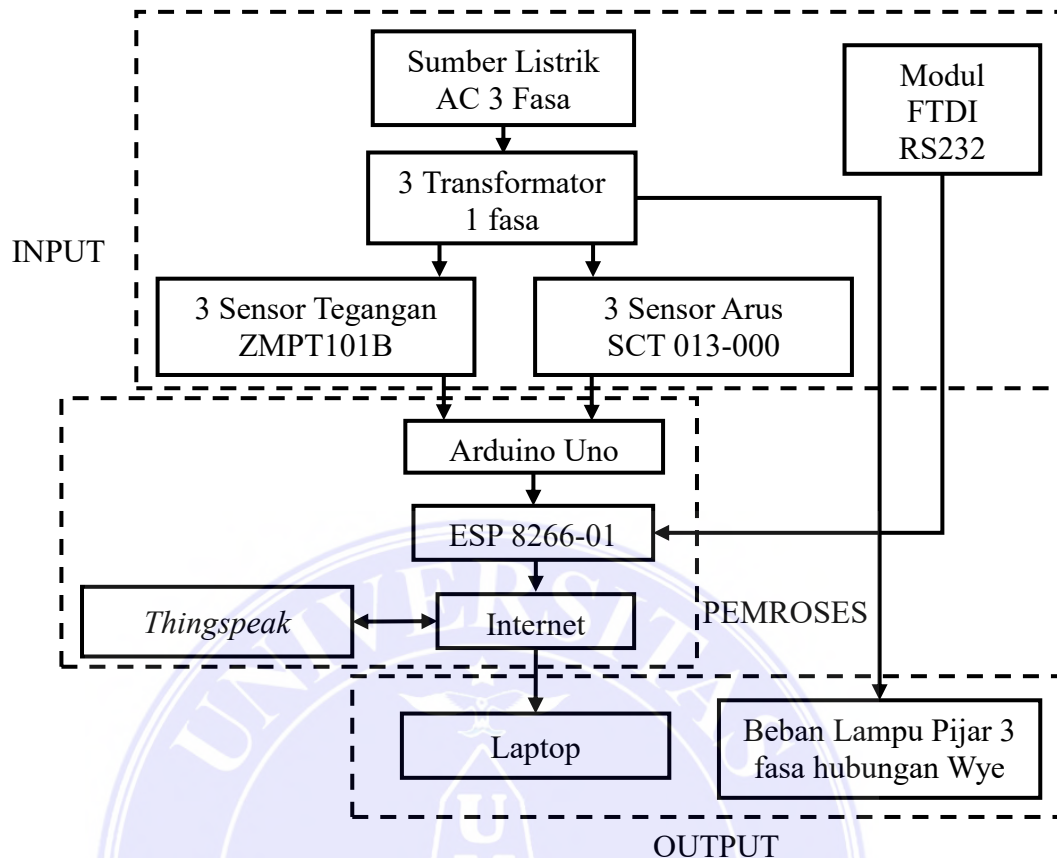
Gambar 3.22. Perangkat simulasi beban

3.2.9. Pengaturan Tegangan Eksitasi Generator

Untuk melakukan pengujian simulasi beban 3 fasa hubungan wye, tegangan fasa ke netral ditentukan pada 220 Volt secara konstan dengan memberikan tegangan eksitasi pada generator menggunakan autotransformator untuk setiap penambahan beban yang dilakukan. Hal ini dapat dilihat pada lampiran 1.

3.2.10. Diagram Keseluruhan Alat

Perancangan perangkat keras bertujuan untuk mendapatkan skema atau gambar rangkaian/perangkat keras dari alat yang akan dibuat. Perancangan perangkat keras dimulai dengan pembuatan blok diagram yang berisi blok-blok penyusun alat serta hubungan fungsional antar blok. Adapun blok diagram dari alat/perangkat keras yang akan dirancang seperti tampak pada Gambar 3.23 berikut:



Gambar 3.23. Blok diagram perancangan sistem

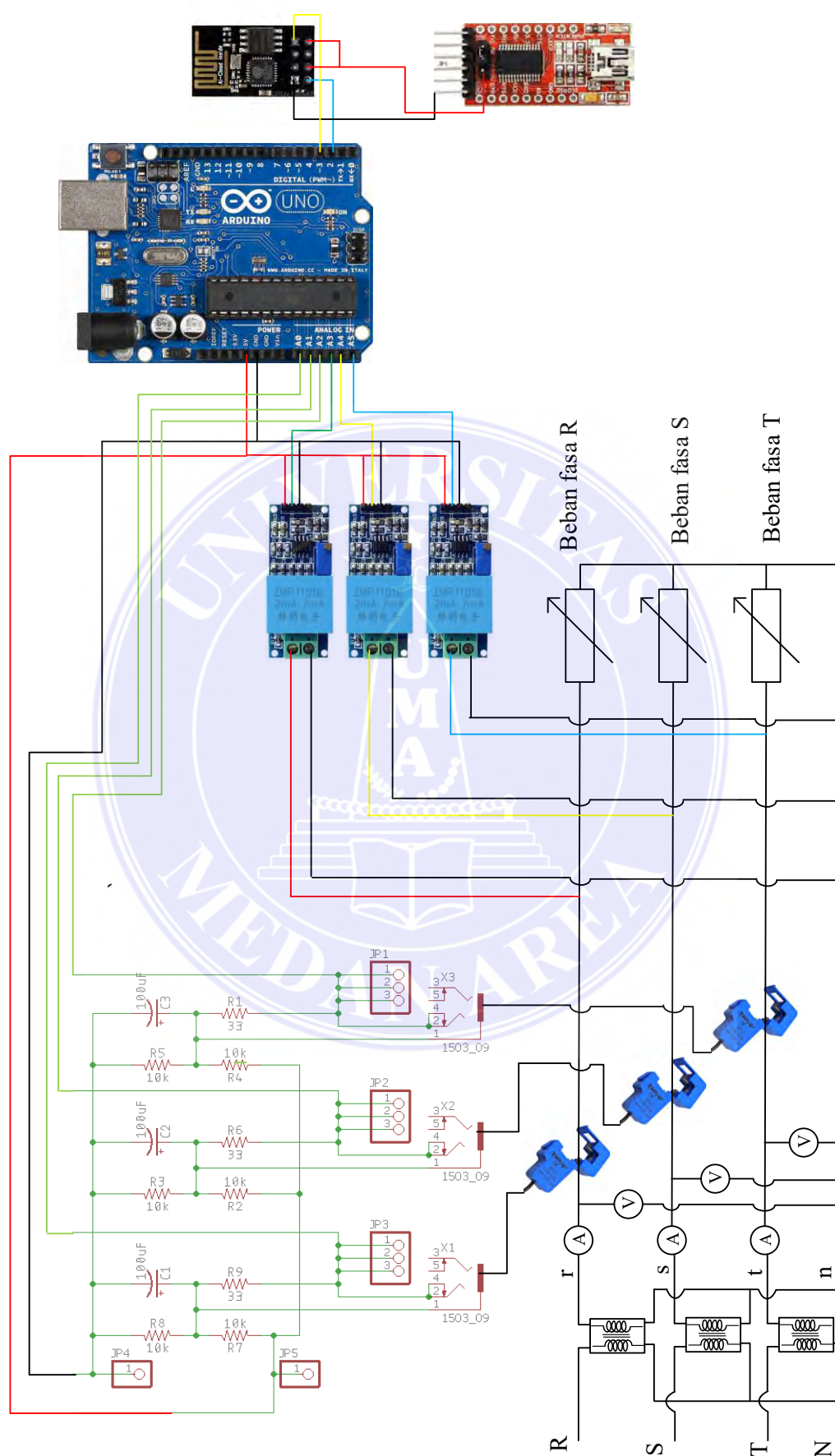
Untuk masing-masing blok dapat dijelaskan sebagai berikut:

- Sumber listrik 3 fasa merupakan perangkat berupa generator 3 fasa yang dikopelkan bersama motor 3 fasa yang ada di Laboratorium Konversi Energi Listrik Program Studi Teknik Elektro Universitas Medan Area dari sumber listrik PLN.
- 3 unit transformator 1 fasa dengan hubungan wye berfungsi memberikan inputan daya listrik ke beban lampu pijar.
- Beban 3 fasa berupa beberapa unit lampu pijar dengan hubungan beban secara wye, merupakan rangkaian dari beberapa unit lampu dengan saklar pada setiap lampu yang digunakan sebagai simulasi ketidakseimbangan beban,

- d. 3 unit sensor arus SCT 013 000 berfungsi mengukur arus fasa R, fasa S dan fasa T pada output dari setiap transformator dengan menjepitkan inti besi pada sensor ke setiap penghantar,
- e. 3 unit sensor tegangan ZMPT 101B berfungsi mengukur tegangan *line to neutral* pada transformator 3 fasa hubungan Wye, yakni R-N, S-N dan T-N,
- f. Arduino Uno merupakan pemroses output dari pengukuran arus oleh SCT 013-000 dan tegangan oleh ZMPT 101B dan selanjutnya mengirimkan data pengukuran ke modul *WiFi*,
- g. ESP8266-01 merupakan modul *WiFi* yang berfungsi mengirimkan data yang diproses Arduino Uno ke *Thingspeak* melalui jaringan internet.
- h. Internet, agar dapat melakukan pengiriman data pengukuran arus dan tegangan ke *thingspeak*, *prototype* menggunakan jaringan internet,
- i. *Thingpeak* merupakan *web server* yang berfungsi media monitoring yang menampilkan data pengukuran yang dikirimkan dari alat dalam bentuk grafik dan dapat diakses melalui komputer maupun smartphone sehingga pengguna dapat mengetahui informasi perubahan arus, tegangan dan persentase ketidakseimbangan beban secara *online*,
- j. Modul FTDI RS232 digunakan sebagai *power supply* eksternal modul *WiFi* ESP8266-01
- k. Laptop digunakan sebagai media monitoring dari hasil pengukuran arus dan tegangan *prototype* ke *web server Thingspeak*.

3.2.11. Rangkaian Keseluruhan

Rangkaian keseluruhan merupakan gabungan dari rangkaian-rangkaian tiap blok yang sudah dibahas sebelumnya, dapat dilihat pada Gambar 3.24 berikut.



Gambar 3.24. Rangkaian keseluruhan sistem

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian pada bab IV, maka dapat ditarik beberapa kesimpulan diantaranya adalah sebagai berikut:

- a. Alat hasil rancangan dapat bekerja melakukan fungsinya sebagai monitoring ketidakseimbangan beban pada *prototype* transformator distribusi dengan mengirimkan data pengukuran sensor arus dan sensor tegangan ke *thingspeak*. Hasil pengukuran tegangan menggunakan sensor tegangan ZMPT 101B sesuai standar *error* alat ukur dan hasil pengukuran arus menggunakan sensor arus SCT 013-000 memerlukan perbaikan akurasi agar memenuhi standar *error* alat ukur.
- b. Informasi pengukuran dari rancangan alat *prototype* ditampilkan dalam bentuk grafik pada *web server Thingspeak* dengan 60 entri data dari 96 entri data keseluruhan dan file dapat di *download* dengan format CSV di dalam *channel thingspeak* yang digunakan.
- c. ESP8266-01 mengirimkan data pengukuran sensor arus dan sensor tegangan yang diolah dari arduino uno ke *Thingspeak* dengan rentang waktu tampilan data 15 detik per entri data via internet. Kecepatan rata-rata transfer data dari *prototype* menuju *thingspeak* adalah 2,825478501 kB/s dengan waktu rata-rata 0,397741s. Data pengukuran yang dikirimkan ke *thingspeak* dapat diakses secara online pada *smartphone* atau PC/Laptop pengguna.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, berikut adalah saran penulis untuk penelitian selanjutnya agar dapat dilakukan pengembangan, ialah:

- a. Dapat melakukan penyempurnaan rangkaian pengkondisi sinyal sensor arus untuk setiap fasa, sehingga memiliki *error* sesuai standar IEC no. 13-B-23,
- b. Menambahkan beban kapasitif dan induktif dalam simulasi ketidakseimbangan beban.



DAFTAR PUSTAKA

- Akbar, R. 2018. Rancang Bangun Alat Monitoring Tegangan, Arus, Daya, Kwh, serta Estimasi Biaya Pemakaian Peralatan Listrik pada Rumah Tangga. Skripsi Universitas Islam Indonesia, Yogyakarta.
- Amaro, N. 2017. Sistem Monitoring Besaran Listrik dengan Teknologi IoT (Internet Of Things). Skripsi Universitas Lampung, Bandar Lampung.
- Artanto, H. 2018. Trainer Iot Berbasis ESP8266 Sebagai Media Pembelajaran Mata Kuliah Komunikasi Data dan Interface di Program Studi Pendidikan Teknik Elektronika UNY. Skripsi Universitas Negeri Yogyakarta, Yogyakarta.
- Iswanto dan Gandi. 2018. Perancangan Dan Implementasi Sistem Kendali Lampu Ruangan Berbasis IoT (Internet Of Things) Android (Studi Kasus Universitas Nurtanio). Jurnal FIKI 9 (1): 38-46.
- Kawihing, A. P., Tuegeh, M., Patras, L. S., & Pakiding, I. M. 2013. Pemerataan Beban Transformator pada Saluran Distribusi Sekunder. E-journal Teknik Elektro dan Komputer.
- Kusuma, Nurul A.A. 2018. Rancang Bangun Smart Home Menggunakan Wemos D1 R2 Arduino Compatible Berbasis ESP8266 ESP-12F. Skripsi Universitas Islam Negeri Syarif Hidayatullah, Jakarta.
- Kurniawan, D. 2018. Alat Deteksi Awal Indikasi Pencurian Listrik pada Kwh Meter Berbasis Mikrokontroler melalui Fasilitas SMS Modem GSM SIM 8001 Berbasis Arduino di PLN Area Madiun. Skripsi Universitas Muhammadiyah Ponorogo, Ponorogo.
- Latupeirissa, L. H. 2017. Pengaruh Ketidakseimbangan Beban Terhadap Arus Netral dan Losses Daya pada Trafo Distribusi Gardu KP-01 Desa Hative Kecil. Jurnal SIMETRIK 7(2): 16–22.
- Nisa, A. 2018. Pemanfaatan Teknologi Internet Of Things Untuk Monitoring Konsentrasi CO dan CO2 Dalam Upaya Mendeteksi Kebakaran Hutan. Skripsi Universitas Hasanuddin, Makassar.
- Nurfaif, M.B. 2017. Rancang Bangun Sistem Rumah Cerdas menggunakan Jaringan Internet. Skripsi Universitas Lampung, Bandar Lampung.
- OpenEnergyMonitor, “CT sensor – Interfacing with an Arduino,” <https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/interface-with-arduino?redirected=true> [Diakses 22 Juni 2020]
- Putra, I.G.P.M.E., dan Darmita, K. I. 2017. Monitoring Penggunaan Daya Listrik sebagai Implementasi Internet Of Things Berbasis ESP8266. SENTRINOV 3: 313-327.

- Rahayu, E. S & Amalia, N. 2018. Perancangan Sistem Informasi “DIAMONS” (Diabetes Monitoring System) Berbasis Internet of Things (IoT). *Jurnal Teknologi* 6(1): 39-51.
- Samsugi, S., Ardiansyah., dan Kastutara, D. 2017. INTERNET OF THINGS (IOT): Sistem Kendali Jarak Jauh Berbasis Arduino Dan Modul Wifi Esp8266. *Rekayasa Teknologi Industri dan Informasi* 295-303
- Sapiie, S., Nishino, O. (2000). *Pengukuran dan Alat-alat Ukur Listrik*. Jakarta: Pradnya Paramita.
- Setiadji, J.S., Machmudsyah, T., dan Isnanto, Y. 2006. Pengaruh Ketidakseimbangan Beban Terhadap Arus Netral Dan Losses Pada Trafo Distribusi. *Jurnal Teknik Elektro* 6(1): 68-73.
- Setiawan, Y., Tanudjaja, H., dan Octaviani, S. 2018. Penggunaan Internet of Things (IoT) untuk Pemantauan dan Pengendalian Sistem Hidroponik. *TESLA* 20(2): 175-182.
- Sugito, Kristian Sandi. 2017. Lampu RGB Alarm Menggunakan ESP-8266. Skripsi Universitas Sanata Dharma, Yogyakarta.
- Tarihoran, S. I. 2017. Sinyal Arus Lebih Pada Kabel ABB Penampang Tertentu. Skripsi Universitas Medan Area, Medan.
- Waworundeng, J dan Lengkong, O. 2018. Sistem Monitoring dan Notifikasi Kualitas Udara dalam Ruangan dengan Platform IoT. *Cogito Smart Journal* 4(1): 94-103.
- Wicaksono, C.A. 2018. Instrumentasi Parameter Daya Listrik Peralatan Listrik Rumah Tangga. Skripsi Universitas Negeri Jakarta, Jakarta.
- Wijaya, M. (2001). *Dasar-dasar Mesin Listrik*. Jakarta: Djambatan.
- Yuliansyah, H. 2016. Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture. *Jurnal Rekayasa dan Teknologi Elektro* 10(2): 68–77.
- Zebua, O., Setiawan, A. H., Soedjarwanto, N., Anggara, J., & Haris, A. 2016. Rancang Bangun Alat Monitoring Ketidakseimbangan Beban Pada Jaringan Tegangan Menengah. *Jurnal Nasional Teknik Elektro* 5(3): 405-413.
- Zikri, A. 2020. Rancang Bangun Sistem Penyiraman Tanaman Otomatis Berbasis Raspberry PI 3 dengan Memanfaatkan *Thingspeak* dan *Interface* Android sebagai Kendali. Skripsi Universitas Islam Negeri Syarif Hidayatullah Jakarta, Jakarta.
- Zuhal. (2000). *Dasar Teknik Tenaga Listrik dan Elektronika Daya*. Jakarta: Gramedia Pustaka Utama.

LAMPIRAN

Lampiran 1. Gambar panel kontrol generator motor Laboratorium Konversi Energi Listrik Program Studi Teknik Elektro UMA.



Gambar panel kontrol generator motor laboratorium Teknik Elektro UMA.

Lampiran 2. *Listing* Program Arduino.

```
#include "EmonLib.h"
#define VOLT_CAL 700.5
EnergyMonitor emon0;
EnergyMonitor emon1;
EnergyMonitor emon2;
EnergyMonitor emon3;
EnergyMonitor emon4;
EnergyMonitor emon5;
#include <SoftwareSerial.h>    //Software Serial library
SoftwareSerial espSerial(2, 3); //Pin 2 and 3 act as TX and RX. Connect them to
TX and RX of ESP8266
#define DEBUG true

String mySSID = "Z97";    // WiFi SSID
String myPWD = "alexander"; // WiFi Password
String myAPI = "27QYJHA547KK9Q4C"; // API Key
String myHOST = "api.thingspeak.com";
String myPORT = "80";
```

```

int pin0 = A0;
int pin1 = A1;
int pin2 = A2;
float a;
float b;
float c;
float d;
float Iavg;
float U;

void setup()
{
  Serial.begin(9600);
  espSerial.begin(115200);
  emon0.current(pin0, 111.1);
  emon1.current(pin1, 111.1);
  emon2.current(pin2, 111.1);
  emon3.voltage(A3, VOLT_CAL, 1.2);
  emon4.voltage(A4, VOLT_CAL, 1.2);
  emon5.voltage(A5, VOLT_CAL, 1.2);

  espData("AT+RST", 1000, DEBUG);           //Reset the ESP8266 module
  espData("AT+CWMODE=1", 1000, DEBUG);       //Set the ESP mode as
station mode
  espData("AT+CWLAP=\"" + mySSID + "\",\"" + myPWD + "\"", 1000, DEBUG);
//Connect to WiFi network
/*while(!esp.find("OK"))
{
  //Wait for connection
}*/
delay(1000);
}

void loop()
{
  float Irms0 = emon0.calcIrms(1480)/1.86;
  float Irms1 = emon1.calcIrms(1480)/1.96;
  float Irms2 = emon2.calcIrms(1480)/1.85;
  Iavg = (Irms0+Irms1+Irms2)/3;
  a = Irms0/Iavg;
  b = Irms1/Iavg;
  c = Irms2/Iavg;
  d = abs(a-1)+abs(b-1)+abs(c-1);
  U = (d*100)/3;
}

```

```

    emon3.calcVI(500,200); //Calculate all. No.of half wavelengths
(crossings),time-out
    emon4.calcVI(500,200);
    emon5.calcVI(500,200);
    float VoltageR = emon3.Vrms; //extract Vrms into Variable
    float VoltageS = emon4.Vrms;
    float VoltageT = emon5.Vrms;

    // Send a random number between 1 and 1000
    String sendData = "GET /update?api_key=";
    sendData += myAPI;
    sendData += "&field1=";
    sendData += String(Irms0);
    sendData += "&field2=";
    sendData += String(Irms1);
    sendData += "&field3=";
    sendData += String(Irms2);
    sendData += "&field4=";
    sendData += String(U);
    sendData += "&field5=";
    sendData += String(VoltageR);
    sendData += "&field6=";
    sendData += String(VoltageS);
    sendData += "&field7=";
    sendData += String(VoltageT);
    sendData += "\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n";
    espData("AT+CIPMUX=1", 1000, DEBUG); //Allow multiple connections
    espData("AT+CIPSTART=0,\"TCP\", \""+ myHOST + "\", "+ myPORT, 1000,
    DEBUG);
    espData("AT+CIPSEND=0," +String(sendData.length()+4),1000,DEBUG);
    espSerial.find(">");
    espSerial.println(sendData);
    Serial.print("IR: ");
    Serial.println(Irms0);
    Serial.print("IS: ");
    Serial.println(Irms1);
    Serial.print("IT: ");
    Serial.println(Irms2);

    espData("AT+CIPCLOSE=0",1000,DEBUG);
    delay(10000);
}

String espData(String command, const int timeout, boolean debug)
{
    Serial.print("AT Command ==> ");
    Serial.print(command);

```

```

Serial.println("  ");

String response = "";
espSerial.println(command);
long int time = millis();
while ( (time + timeout) > millis())
{
    while (espSerial.available())
    {
        char c = espSerial.read();
        response += c;
    }
}
if (debug)
{
    //Serial.print(response);
}
return response;
}

```

LAMPIRAN 3. Hasil pengiriman data pengukuran *prototype* ke *thingspeak*

WAKTU	entry_id	R (A)	S (A)	T (A)	Unbalance (%)	R-N (V)	S-N (V)	T-N (V)
2020-08-10 12:28:20 WIB	1,0	0.23	0.23	0.27	6.63	2.62	2.54	2.77
2020-08-10 12:28:36 WIB	2,0	0.20	0.21	0.27	11.83	2.50	2.59	2.67
2020-08-10 12:28:52 WIB	3,0	0.20	0.23	0.27	10.26	2.30	2.49	2.70
2020-08-10 12:29:09 WIB	4,0	0.22	0.25	0.27	7.23	2.50	2.63	2.91
2020-08-10 12:29:25 WIB	5,0	0.22	0.25	0.27	7.23	2.50	2.62	2.55
2020-08-10 12:29:42 WIB	6,0	0.21	0.23	0.27	9.37	2.32	2.54	2.74
2020-08-10 12:29:58 WIB	7,0	0.21	0.24	0.27	8.89	2.46	2.52	2.82
2020-08-10 12:30:15 WIB	8,0	0.24	0.24	0.27	5.81	2.65	2.79	2.67
2020-08-10 12:30:31 WIB	9,0	0.24	0.24	0.27	5.33	2.25	2.56	2.43
2020-08-10 12:30:47 WIB	10,0	0.21	0.24	0.27	8.71	2.37	2.50	3.05
2020-08-10 12:31:04 WIB	11,0	0.73	0.60	0.63	7.66	130.60	135.59	133.76
2020-08-10 12:31:20 WIB	12,0	0.76	0.82	0.93	7.12	220.86	219.65	217.55
2020-08-10 12:31:37 WIB	13,0	0.74	0.91	0.89	8.46	221.51	219.45	219.15
2020-08-10 12:31:53 WIB	14,0	0.75	0.95	0.76	10.46	221.64	221.48	220.37
2020-08-10 12:32:10 WIB	15,0	0.83	0.93	0.72	8.89	220.10	222.24	220.22
2020-08-10 12:32:26 WIB	16,0	0.88	0.87	0.68	10.76	218.60	222.03	218.51
2020-08-10 12:32:42 WIB	17,0	0.94	0.76	0.79	8.91	219.00	223.65	215.50
2020-08-10 12:32:59 WIB	18,0	0.87	0.72	0.92	8.93	220.25	224.30	216.88
2020-08-10 12:33:15 WIB	19,0	0.79	0.77	0.96	9.57	220.82	222.12	217.10
2020-08-10 12:33:32 WIB	20,0	0.72	0.88	0.91	9.39	221.82	219.57	218.57

LAMPIRAN 3. Hasil pengiriman data pengukuran *prototype* ke *thingspeak*

2020-08-10 12:33:48 WIB	21,0	0.76	0.94	0.77	9.48	221.08	220.26	219.12
2020-08-10 12:34:05 WIB	22,0	0.83	0.95	0.70	10.55	220.07	221.99	219.70
2020-08-10 12:34:21 WIB	23,0	0.88	0.89	0.68	11.30	220.69	224.13	216.88
2020-08-10 12:34:38 WIB	24,0	0.92	0.83	0.71	8.83	217.63	222.07	215.94
2020-08-10 12:34:54 WIB	25,0	0.92	0.75	0.81	7.27	219.77	224.33	215.22
2020-08-10 12:35:10 WIB	26,0	0.84	0.72	0.92	8.66	220.98	224.31	215.12
2020-08-10 12:35:27 WIB	27,0	0.78	0.78	0.94	8.82	223.10	223.24	215.64
2020-08-10 12:35:43 WIB	28,0	0.73	0.88	0.93	8.94	221.39	219.62	218.53
2020-08-10 12:36:16 WIB	29,0	0.90	1.53	1.45	20.39	216.31	215.33	202.97
2020-08-10 12:36:32 WIB	30,0	0.95	1.56	1.50	19.50	225.76	225.08	212.13
2020-08-10 12:36:49 WIB	31,0	0.97	1.61	1.42	18.22	225.58	224.99	213.10
2020-08-10 12:37:05 WIB	32,0	0.95	1.57	1.46	18.90	224.50	223.94	211.09
2020-08-10 12:37:22 WIB	33,0	0.93	1.54	1.52	20.21	227.23	226.26	214.01
2020-08-10 12:37:38 WIB	34,0	0.94	1.55	1.51	19.78	226.13	226.01	212.47
2020-08-10 12:37:55 WIB	35,0	0.96	1.55	1.49	18.80	228.77	226.95	213.73
2020-08-10 12:38:11 WIB	36,0	0.96	1.55	1.49	18.77	226.11	226.89	213.23
2020-08-10 12:38:27 WIB	37,0	0.92	1.60	1.48	20.66	226.00	225.27	212.55
2020-08-10 12:38:44 WIB	38,0	0.95	1.61	1.44	19.33	225.97	225.34	213.39
2020-08-10 12:39:00 WIB	39,0	0.93	1.65	1.45	20.31	226.68	225.60	215.04
2020-08-10 12:39:17 WIB	40,0	0.94	1.67	1.42	20.05	228.65	228.06	216.66
2020-08-10 12:39:33 WIB	41,0	0.96	1.65	1.43	19.26	227.32	226.05	215.05
2020-08-10 12:39:50 WIB	42,0	0.92	1.68	1.43	20.98	231.45	228.93	216.08
2020-08-10 12:40:06 WIB	43,0	0.92	1.69	1.42	21.09	228.33	225.35	217.96
2020-08-10 12:40:22 WIB	44,0	1.51	1.73	2.14	13.04	218.18	220.03	209.35
2020-08-10 12:40:39 WIB	45,0	1.56	1.74	2.13	11.63	223.03	227.49	215.58
2020-08-10 12:40:55 WIB	46,0	1.52	1.75	2.16	12.85	222.53	224.63	213.74
2020-08-10 12:41:12 WIB	47,0	1.49	1.71	2.24	15.71	221.99	225.43	212.89
2020-08-10 12:41:28 WIB	48,0	1.48	1.65	2.26	17.21	221.55	225.58	212.32
2020-08-10 12:41:44 WIB	49,0	1.49	1.63	2.26	17.35	219.23	223.51	210.30
2020-08-10 12:42:50 WIB	50,0	1.51	1.60	2.27	17.67	223.11	226.32	211.55
2020-08-10 12:43:07 WIB	51,0	1.54	1.56	2.25	17.27	219.53	226.80	211.00
2020-08-10 12:43:23 WIB	52,0	2.18	2.42	2.88	10.27	205.82	208.01	195.84
2020-08-10 12:43:40 WIB	53,0	2.41	2.34	2.96	10.14	219.62	228.59	212.67
2020-08-10 12:43:56 WIB	54,0	2.42	2.34	2.96	10.03	221.30	230.25	213.01
2020-08-10 12:44:12 WIB	55,0	2.39	2.37	2.99	10.53	221.71	230.79	213.59
2020-08-10 12:44:29 WIB	56,0	2.42	2.34	2.98	10.40	222.93	230.58	214.04
2020-08-10 12:44:45 WIB	57,0	2.45	2.37	2.92	8.82	223.43	230.40	214.71
2020-08-10 12:45:02 WIB	58,0	2.48	2.43	2.86	6.93	222.62	230.27	215.57
2020-08-10 12:45:18 WIB	59,0	2.49	2.45	2.82	6.03	223.17	229.72	216.74
2020-08-10 12:45:34 WIB	60,0	2.46	2.52	2.85	6.17	223.03	229.05	216.31
2020-08-10 12:45:51 WIB	61,0	2.38	2.57	2.93	7.67	226.05	229.08	215.42

LAMPIRAN 3. Hasil pengiriman data pengukuran *prototype* ke *thingspeak*

2020-08-10 12:46:07 WIB	62,0	2.29	2.53	3.03	10.44	225.51	228.59	214.01
2020-08-10 12:46:24 WIB	63,0	2.27	2.50	3.02	10.90	221.03	224.97	209.94
2020-08-10 12:46:40 WIB	64,0	2.26	2.52	3.02	10.75	222.21	225.93	211.05
2020-08-10 12:46:56 WIB	65,0	2.25	2.52	3.03	10.97	223.45	226.98	212.01
2020-08-10 12:47:13 WIB	66,0	2.26	3.18	2.99	13.02	223.79	217.47	206.35
2020-08-10 12:47:30 WIB	67,0	2.34	3.30	3.71	16.57	226.54	224.61	202.35
2020-08-10 12:48:02 WIB	68,0	2.51	3.23	3.52	12.44	226.77	225.54	208.34
2020-08-10 12:48:19 WIB	69,0	2.30	3.21	3.78	17.12	229.67	229.93	206.29
2020-08-10 12:48:35 WIB	70,0	2.48	3.31	3.57	13.66	229.54	226.41	208.68
2020-08-10 12:48:51 WIB	71,0	2.33	3.15	3.77	16.28	227.24	229.01	205.12
2020-08-10 12:49:08 WIB	72,0	2.48	3.29	3.55	13.38	227.74	225.63	208.56
2020-08-10 12:49:24 WIB	73,0	2.31	3.20	3.77	16.92	228.00	229.48	205.27
2020-08-10 12:49:41 WIB	74,0	2.51	3.20	3.54	12.33	225.62	225.68	207.77
2020-08-10 12:49:57 WIB	75,0	2.31	3.27	3.72	17.03	228.26	225.25	204.36
2020-08-10 12:50:13 WIB	76,0	2.38	3.78	3.60	17.92	223.02	213.20	198.83
2020-08-10 12:50:30 WIB	77,0	2.30	3.89	3.71	20.18	224.34	212.54	198.90
2020-08-10 12:50:46 WIB	78,0	3.19	3.90	3.73	7.66	228.62	224.47	216.18
2020-08-10 12:51:03 WIB	79,0	3.28	4.04	3.66	6.95	232.29	223.93	216.26
2020-08-10 12:51:19 WIB	80,0	3.16	4.06	3.82	9.35	231.80	226.81	214.46
2020-08-10 12:51:36 WIB	81,0	3.22	3.91	3.69	7.11	228.62	224.06	216.51
2020-08-10 12:51:52 WIB	82,0	3.22	4.10	3.76	8.60	231.87	223.36	214.04
2020-08-10 12:52:08 WIB	83,0	3.13	3.91	3.79	8.91	230.08	226.50	216.23
2020-08-10 12:52:25 WIB	84,0	3.29	4.07	3.68	7.10	232.84	224.57	216.50
2020-08-10 12:52:41 WIB	85,0	3.14	3.95	3.83	9.14	231.41	227.06	216.06
2020-08-10 12:52:58 WIB	86,0	3.84	4.00	3.79	2.13	222.89	224.37	218.73
2020-08-10 12:53:14 WIB	87,0	3.94	3.96	3.77	2.00	227.14	228.71	226.02
2020-08-10 12:53:31 WIB	88,0	3.93	4.12	3.91	2.28	229.46	230.00	222.80
2020-08-10 12:53:47 WIB	89,0	3.99	3.94	3.77	2.24	227.10	227.79	225.24
2020-08-10 12:54:03 WIB	90,0	3.98	4.10	3.89	1.88	228.55	227.89	221.51
2020-08-10 12:54:20 WIB	91,0	3.93	3.93	3.79	1.61	226.55	228.24	225.71
2020-08-10 12:54:36 WIB	92,0	3.94	4.11	3.89	2.14	227.99	228.81	222.27
2020-08-10 12:54:53 WIB	93,0	4.01	3.97	3.72	3.08	225.70	226.44	223.53
2020-08-10 12:55:09 WIB	94,0	3.92	4.12	3.90	2.33	228.57	230.10	222.71
2020-08-10 12:55:26 WIB	95,0	0.25	0.26	0.27	2.20	3.25	3.08	3.42
2020-08-10 12:55:42 WIB	96,0	0.24	0.26	0.27	4.03	2.79	3.05	3.24