

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan listrik khususnya mobil listrik pada beberapa tahun terakhir ini banyak dikembangkan sebagai upaya penghematan bahan bakar fosil. Ketersediaan cadangan fosil tentunya semakin berkurang seiring semakin banyaknya kendaraan konvensional yang digunakan. Dengan kendaraan listrik, maka sumber energinya berasal dari energi listrik yang termasuk dalam kategori energi *renewable*.

Mobil listrik memerlukan sebuah stasiun pengisian energi listrik layaknya SPBU bagi kendaraan konvensional. SPBU bagi kendaraan listrik disebut dengan *charging station*. *Charging station* dapat memiliki *hybrid power source* yang berarti sumber listriknya berasal lebih dari satu titik, yang paling umum berasal dari solar panel dan PLN, contohnya pada rencana pembangunan sistem catudaya hibrida dalam rangka pengembangan Mobil Listrik Nasional (MOLINA). Dalam proses pengisian baterai perlu adanya monitoring tegangan, arus, maupun daya dari konsumsi *charging station* tersebut, agar dapat diketahui efisiensi dan kondisi di lapangan dari *charging station* tersebut oleh teknisi. Manfaat lebih lanjut dengan diketahuinya nilai dari arus dan tegangan ialah biaya pemakaian daya listrik untuk melakukan proses pengisian daya mobil listrik maupun bus listrik dapat diketahui.

Agar proses monitoring dapat dilakukan dari jarak jauh tanpa harus mendatangi *plant*, maka diperlukan sebuah sistem monitoring yang bersifat tele-monitoring. Oleh karena itu pada tugas akhir ini telah dirancang sebuah sistem monitoring tegangan dan arus pada *charging station* berbasis web. Dengan menggunakan web server, maka *client* yang terhubung dengan jaringan intranet pada sistem monitoring tersebut cukup menggunakan *web-browser* untuk memonitoring proses *charging* baterai secara *real time*. Pada sistem monitoring ini juga terdapat *history* atau data *logger* dari nilai-nilai sebelumnya. Hasil dari tugas akhir ini diharapkan mampu memberi kemudahan bagi admin *charging station* dalam memonitoring keadaan dilapangan dan tidak menutup kemungkinan untuk dikembangkan lebih lanjut sebagai kontrol jarak jauh bagi *charging station* tersebut.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah :

1. Bagaimana performa sensor terkait pengukuran tegangan tinggi dan arus tinggi.
2. Bagaimana memproses data analog dari sensor agar menjadi data digital yang merepresentasikan nilai sebenarnya dari arus dan tegangan yang diukur.
3. Bagaimana proses pengiriman data dari mikrokontroler menuju database server dapat berlangsung.
4. Bagaimana tampilan sistem monitoring mode *real time* dan data *logger*.

1.3 Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut :

1. Mengetahui respon dari sensor tegangan dan arus rating tinggi.
2. Memperoleh teknik yang tepat untuk melakukan pembacaan nilai tegangan dan arus dengan rating tinggi.
3. Mewujudkan sebuah sistem monitoring jarak jauh yang diimplementasikan pada pembangunan *charging station* MOLINA di ITS untuk memonitoring tegangan dan arus di titik tertentu.

1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Sensor tegangan yang digunakan ialah LEM LV 25-P.
2. Sensor arus yang digunakan ialah LEM LA 55-P.
3. Tegangan DC yang diukur tidak lebih dari 500 V.
4. Arus yang diukur tidak lebih dari 50 A.
5. Mikrokontroler yang digunakan ialah arduino Uno R3 dengan Ethernet *shield* sebagai modul untuk terhubung dengan jaringan.
6. Router yang digunakan untuk membuat jaringan lokal ialah TP-LINK TL-MR3020.
7. Data arus dan tegangan disimpan dalam Database MySQL dan dengan webserver Apache di OS Windows 7 64-bit.
8. Hanya 1000 data terakhir yang disimpan dalam Database MySQL.

1.5 Metodologi Penelitian

Dalam penyelesaian Tugas Akhir ini digunakan metodologi sebagai berikut :

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, *proceeding*, dan artikel-artikel di internet. Meliputi :

- Pemilihan sensor tegangan dan arus rating tinggi dengan standar industri.
- Pengkondisian sinyal output dari sensor agar dapat dibaca oleh mikrokontroler.
- Mempelajari database MySQL serta pemrograman Arduino, PHP dan HTML.

2. Perancangan Hardware

Pada tahap ini dirancang sebuah hardware yang terdiri dari rangkaian sensor tegangan, rangkaian sensor arus, *powersupply +/-, adder*, dan sistem mikrokontroler. Hasil pengukuran akan ditampilkan di web, dan database pengukuran dalam bentuk tabel dan grafik.

3. Perancangan Software

Pada tahap ini dibuat program untuk akuisisi data menggunakan IDE Arduino ERW 1.0.5, sedangkan untuk keperluan monitoring menggunakan pemrograman PHP, HTML dan database MySQL.

4. Pengujian Sistem

Pengujian sistem dilakukan dengan menguji data pembacaan sensor, proses linearisasi, dan mengintegrasikannya pada sistem agar data tersebut dapat tampil pada browser dan juga tersimpan pada database.

5. Penulisan Laporan Akhir

Tahap penulisan laporan Tugas Akhir dilakukan setelah mendapatkan data – data yang dibutuhkan.

1.6 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

➤ Bab 1 : Pendahuluan

Bab ini meliputi latar belakang, perumusan masalah, tujuan, sistematika penulisan, metodologi, dan relevansi.

➤ Bab 2 : Dasar Teori

Bab ini menjelaskan tentang berbagai macam teori-teori penunjang dalam pengerjaan tugas akhir ini, diantaranya : sensor tegangan dan

arus yang digunakan, pengkondisian sinyal output sensor, mikrokontroler, dan teori tentang webserver dan database.

➤ **Bab 3: Perancangan Sistem**

Pada bab ini menjelaskan tentang langkah-langkah dalam merencanakan dan membangun sistem secara keseluruhan baik *software* maupun *hardware* agar didapatkan sistem monitoring charging station kendaraan listrik berbasis web.

➤ **Bab 4 : Pengujian dan Analisis**

Bab ini menjelaskan data yang didapat dari pengujian keseluruhan sistem beserta analisisnya.

➤ **Bab 5 : Penutup**

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangannya.

1.7 Relevansi

Hasil dari tugas akhir ini diharapkan mampu membantu *user/admin* charging station dalam memonitoring besar arus dan tegangan dilapangan, tanpa harus ke lapangan untuk mengukurnya secara manual. Pengimplementasian mikrokontroler di sini sebagai interface antara sensor dan komputer server. Digunakannya *database MySql* sebagai penyimpan data pada komputer server dengan maksud agar sistem ini menjadi sistem monitoring dengan mode *real time* dan data *logger*.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

Tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan teknologi yang telah ada. Sedangkan bagian dasar teori menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem pada tugas akhir ini.

2.1 Tinjauan Pustaka

2.1.1 *Electric Vehicle (EV) Charging Station*

Electric Vehicle (EV) charging stations merupakan sumber external *charger* yang diperuntukkan bagi kendaraan listrik. Setiap *station* mempunyai *jack / plug* yang dihubungkan ke kendaraan listrik, yang kemudian menyuplainya dengan arus listrik yang dibutuhkan untuk melakukan pengisian daya.

Di Amerika, Eropa dan Jepang menerapkan *charging station* komersial sesuai standard SAE J1772 *socket*. Semua *EV* terbaru pabrikan seperti Chevrolet Volt, Mitsubishi i-MiEV dan Nissan LEAF kompatibel dengan *socket* J1772. Standar yang lain ialah CHAdeMO, yang hanya bisa di implementasikan untuk DC *fast charging station*[1].

Ada tiga level/mode pengisian pada *charging station* sesuai dengan outputnya, yaitu Level 1, Level 2, dan CHAdeMO. Perbandingannya terlihat pada tabel 2.1.



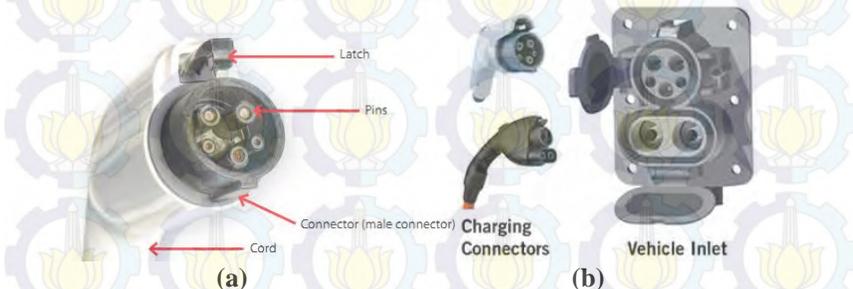
Gambar 2.1 *Charging station*[2]

Tabel 2.1 Perbandingan level/mode pengisian[1]

CHARACTERISTIC	LEVEL 1	LEVEL 2	CHAdEMO
Voltage	120 V	208 or 240 V	Up to 500 V
Type of current	AC	AC	DC
Typical output	1.4 kW	3.6 or 7.2 kW	50 kW
Maximum output	1.9 kW	19.2 kW	50 kW
Charge time ^a	12 h ^a	2.5-5 h ^a	20 min ^a
Connector	J1772	J1772	CHAdEMO

- Waktu pengisian untuk baterai 16-kWh yang habis ter-*discharge*.
- 80% dari kapasitas maksimum (12.8 kWh). Tidak diperkenankan melakukan pengisian penuh pada standard CHAdEMO.

Untuk *charging station* level 1 dan level 2 sama-sama memiliki tipe output arus AC, dan konektor yang sama yaitu J1772, tetapi tegangan dan daya yang mampu disuplai oleh *charging station* level 2 lebih tinggi. Sedangkan level CHAdEMO sangat berbeda yaitu outputnya berupa arus DC. Daya yang dapat disuplai oleh *charging station* level CHAdEMO jauh lebih tinggi daripada level 1 dan 2, oleh karena itu level/mode ini disebut *DC Fast Charge*. *DC Fast Charge* berbeda dengan 2 level sebelumnya dalam hal teknis, yaitu komponen *charger* berada di *charging station*, sedangkan level 1 dan 2 komponen *chargernya* sudah *on-board EV*. Jadi *DC Fast Charge* menyalurkan daya DC secara langsung ke baterai kendaraan listrik. Terlepas dari semua itu, semua *EV* memiliki kabel portable untuk melakukan pengisian daya melalui terminal listrik konvensional (120/220V AC)[1].

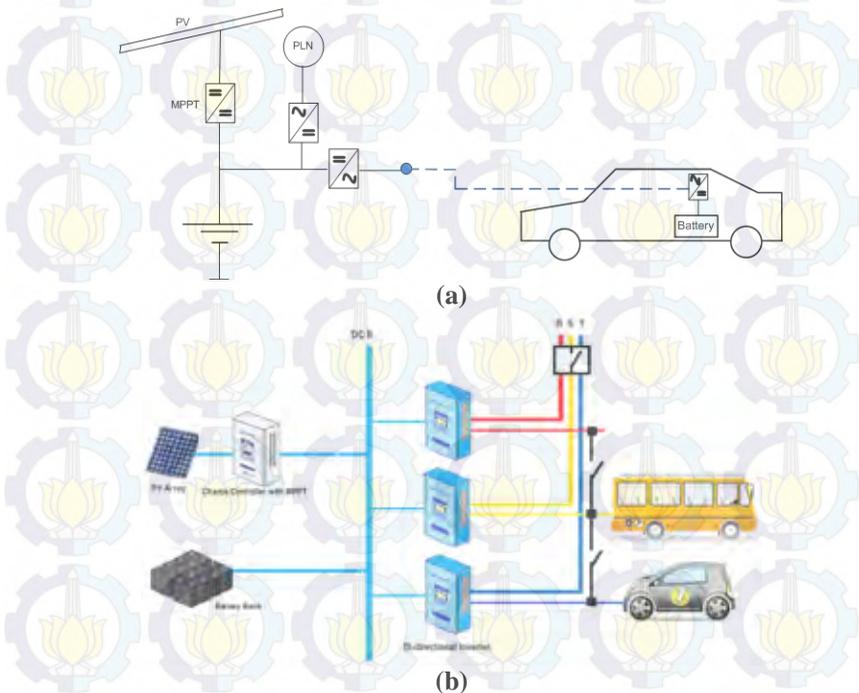


Gambar 2.2 (a) Konektor J1772[1] (b) Konektor J1772 yang ditambah 2 pin lagi agar kompatibel untuk *DC Fast Charge*[3]

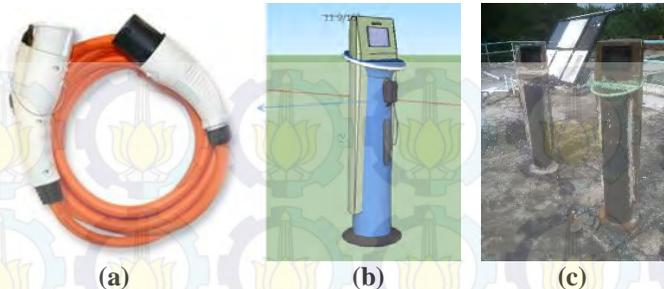
2.1.2 Rencana Pembangunan *Hybrid Power Source EV Charging Station* MOLINA ITS

Charging station dapat memiliki lebih dari satu sumber energi listriknya. Umumnya sumber energi tersebut berasal dari listrik jala-jala, atau di Indonesia bisa berasal dari PLN. Tetapi melihat isu lingkungan terkini, maka sumber energi listrik sebisa mungkin berasal dari energi *renewable*, contohnya sinar matahari. Maka dari itu, *hybrid charging station* yang dimaksud di sini memiliki dua sumber energinya, dari listrik jala-jala dan dari *PV (Photovoltaic)*.

Di ITS, infrastruktur *charging station* terkait dengan proyek MOLINA (Mobil Listrik Nasional) sedang dalam proses riset dan pembuatan *prototype*. Sumber energi yang digunakan berasal dari PLN dan *PV*. Direncanakan terdapat 3 *slot* untuk pengisian kendaraan listrik, yaitu 2 untuk *electric car (EC)*, dan 1 untuk *electric bus (EB)*[4].



Gambar 2.3 (a) Skema *charging station hybrid* [4](b) Skema *charging station MOLINA*[4]



Gambar 2.4 (a) Konektor SAE J1772[4] (b) Desain *charging transaction*[4] (c) Prototype *charging transaction*

Charging station MOLINA ini akan menggunakan standard SAE J1772 dan *charging transaction*nya didesain seperti pada gambar 2.4 (b). Untuk karakteristik *charging*nya seperti berikut[4] :

- Mobil Listrik :
 - *Slow Charging* :
 - Arus Pengisian : 6 A @ 220 VAC
 - Kapasitas daya : 1.320 Watt
 - Waktu pengisian : 10 jam
 - *Fast Charging* :
 - Arus Pengisian : 11 A @ 220 VAC
 - Kapasitas daya : 2.420 Watt
 - Waktu pengisian : 5-6 jam
- Bus Listrik :
 - Arus Pengisian : 50 A @ 220 VAC
 - Kapasitas daya : 11.000 Watt
 - Waktu pengisian : 10 jam

2.1.3 WebDAQ/100 Data Acquisition System

DAQ adalah singkatan untuk " *Data Acquisition* ". WebDAQ otomatis dapat merekam, plot dan ekspor data digital atau analog, di kontrol *remotely* melalui halaman web, *command line session* atau antarmuka perangkat lunak dengan menggunakan program spreadsheet atau lab seperti DASYlab, LabView atau Mathematica[5]. Setiap webDAQ/100 berisi sebuah komputer lengkap dengan sebuah web server di dalam *box*. Dengan mengkoneksikannya ke sebuah jaringan atau sambungan langsung dengan ethernet *card* komputer, maka kontrol

secara penuh dapat dilakukan dengan membuka web *browser* dan mengetikkan alamat IP dari webDAQ/100[6]. Harga dari datalogger ini berada dikisaran \$1295[7]. Spesifikasi lengkap dari webDAQ sebagai berikut :

- 32 channels 12-bit A/D input
- 12 single-ended thermocouple inputs with built in cold junction compensation sensor
- 8 channels 10-bit D/A output
- 4 digital output
- 4 digital input
- Ethernet and serial ports



Gambar 2.5 webDAQ/100 Data Acquisition System[6]



Gambar 2.6 webDAQ/100 home page[6]

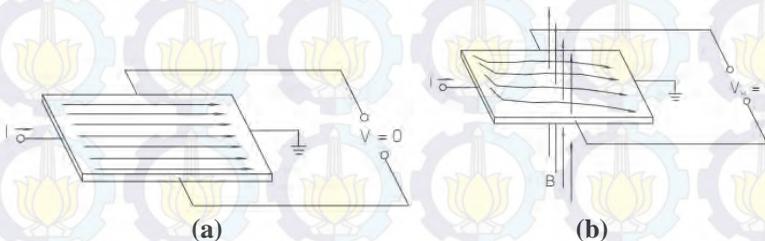
2.2 Sensor Arus LA 55-P

LA 55-P merupakan sensor arus yang merupakan produk dari LEM. Sensor arus LA 55-P dapat digunakan untuk mengukur arus DC, AC, pulsa, dan memiliki *galvanic isolation* diantara *primary circuit* (berhubungan dengan arus tinggi) dan *secondary circuit* (rangkaiannya elektronik). Dengan arus nominal yang dapat diukur sebesar 50 A.

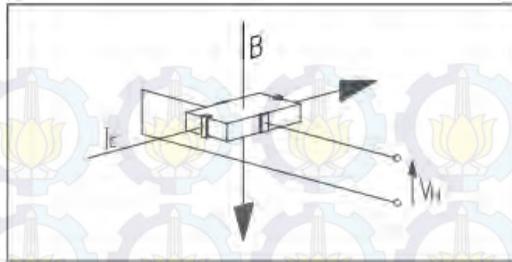
Berdimensi 3.6 cm x 2.7 cm x 1.4 cm (panjang x lebar x tinggi) dan memiliki 3 pin yang ukurannya standar *pcb mounting*. Sumber tegangannya $\pm 12..15$ V. Prinsip kerja dari sensor ini mengimplementasi teori *hall effect* yang sistemnya *closed loop*.

2.2.1 Teknologi Hall Effect LEM

Hall effect ditemukan oleh Dr. Edwin Hall pada tahun 1879 ketika menjalani pendidikan doktor di John Hopkins University Baltimore. Dr. Hall menemukan bahwa ketika medan magnet ditempatkan memotong tegak lurus permukaan datar dari lempengan segi empat emas yang dialiri arus dari salah satu tepi, timbul perbedaan potensial pada tepi yang lain[8]. Perbedaan potensial ini timbul dikarenakan adanya gaya Lorentz, $FL = q \cdot (V \times B)$, yang berlaku pada muatan yang bergerak menembus medan magnet[9]. Terlihat pada gambar 2.7 (a) pada kondisi ideal, jika tidak ada medan magnet, maka tidak ada tegangan pada tepinya. Tetapi jika ada medan magnet seperti ditunjukkan pada gambar 2.7 (b), maka arus akan membelok sesuai kaidah Lorentz sehingga menyebabkan elektron berkumpul pada salah satu tepi. Hal ini membuat perbedaan jumlah elektron diantara tepi, sehingga timbul perbedaan potensial, yang selanjutnya disebut tegangan Hall (V_{Hall}). Pada sensor arus dari LEM, LEM merepresentasikan parameter elektrik dari *hall effect* seperti pada gambar 2.8.



Gambar 2.7 Prinsip *hall effect*, (a) tanpa medan magnet[8] (b) adanya medan magnet[8]



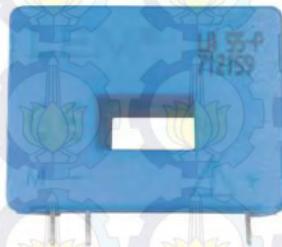
Gambar 2.8 Representasi parameter elektrik dari *hall effect*[9]

Dengan konfigurasi seperti gambar 2.8, maka tegangan hall dari sensor ini dirumuskan :

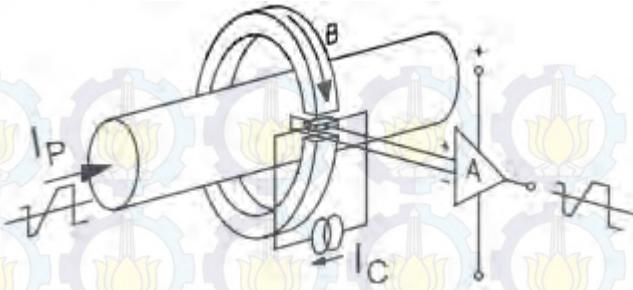
$$V_H = K / d \cdot I_C \cdot B + V_{OH} \quad (1)$$

Dimana ‘K’ adalah konstanta Hall dari material konduktor, ‘d’ adalah ketebalan lempengan konduktor, dan ‘V_{OH}’ adalah tegangan *offset* dari generator hall dalam keadaan tanpa adanya medan magnet. Generator hall adalah keseluruhan konfigurasi seperti yang disebutkan diatas, dan untuk ‘K / d • I_C’ disebut sensitivitas generator Hall[9].

Pada gambar 2.10 terlihat rangkaian skema perubahan dari arus utama yang diukur menjadi tegangan output, yang disebut sebagai *open loop hall effect current transducer*. Dengan begitu, kerapatan flux magnet, B, proporsional terhadap arus primer, I_P, dan tegangan Hall, V_H, proporsional terhadap kerapatan flux. Sehingga output dari generator Hall akan proporsional terhadap arus primer, I_P, ditambah tegangan offset V_{OH}.



Gambar 2.9 Sensor arus LA 55-P[10]

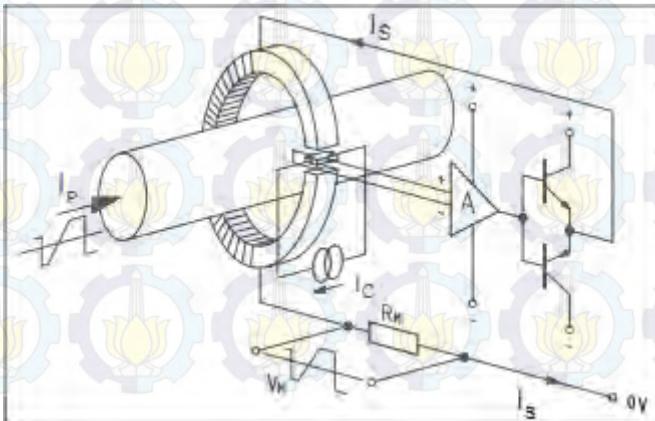


Gambar 2.10 Konversi dari arus primer (I_P) menjadi tegangan output[9]

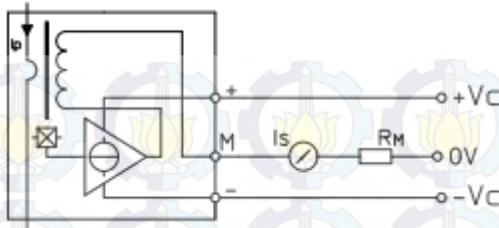
2.2.2 Closed Loop Hall Effect Current Transducer

Open loop current transducer menguatkan tegangan generator Hall untuk mendapatkan tegangan output, sedangkan pada *closed loop transducer* menggunakan tegangan generator Hall untuk membuat arus kompensasi (gambar 2.11) pada kumparan sekunder untuk menciptakan flux total, yang besarnya nol. Dengan kata lain, arus sekunder, I_s , menimbulkan flux yang amplitudonya sama, tetapi arahnya berbeda, terhadap flux yang ditimbulkan oleh arus primer[9].

Mengoperasikan generator Hall pada kondisi zero flux akan mengeliminasi penguatan yang tidak diinginkan dari pengaruh suhu. Keuntungan tambahan dari konfigurasi ini ialah kumparan sekunder akan berfungsi sebagai current transformer pada frekuensi yang lebih tinggi,



Gambar 2.11 Prinsip kerja dari *closed loop transducer*[9]



Gambar 2.12 Rangkaian pengukuran dasar LA 55-P[9]

yang secara signifikan akan menambah bandwidth dan mengurangi *response time* dari transducer. Ketika flux magnet telah terkompensasi secara penuh (*zero*), potensial magnetik (ampere-turns) dari dua koil adalah identik, karenanya:

$$N_P \cdot I_P = N_S \cdot I_S$$

$$\text{yang juga bisa ditulis } I_S = I_P \cdot N_P / N_S \quad (2)$$

Dengan begitu, arus sekunder, I_S , merupakan duplikat dari bentuk arus primer, I_P . Dengan memberi resistor R_M secara seri dengan kumparan sekunder, maka menimbulkan tegangan output yang bentuknya seperti arus primer, I_P . N_S bernilai 1000 dan I_S nominal bernilai 50 mA (50 A / 1000)[9].

2.2.3 Rangkaian Pengukuran Dasar

Sensor LA 55-P memiliki 3 pin seperti tampak pada gambar 2.7 yaitu pin +, M, dan -. Gambar 2.12 menunjukkan rangkaian pengukuran dasar sensor LA 55-P. Sesuai dengan *datasheet*, pemasangan R_M bernilai 10 – 100 Ohm pada suhu hingga 70°C, dengan sumber tegangan +V_C = 12V dan -V_C = -12V.

2.3 Sensor tegangan LV 25-P

LV 25-P merupakan sensor tegangan yang merupakan produk dari LEM. Sensor arus LA 55-P dapat digunakan untuk mengukur tegangan DC, AC, pulsa, dan memiliki *galvanic isolation* diantara *primary circuit* (berhubungan dengan tegangan tinggi) dan *secondary circuit* (rangkaiannya elektronik). Dengan range tegangan nominal yang dapat diukur sebesar 10 .. 500 V. Berdimensi 2.9 cm x 2.6 cm x 1.5 cm (panjang x lebar x tinggi) dan memiliki 5 pin yang ukurannya standar *pcb mounting*. Sumber tegangannya +12..15 V. Prinsip kerja dari sensor ini sama dengan sensor arus LA 55-P yaitu mengimplementasi teori *hall effect* yang sistemnya *closed loop*. Hanya berbeda pada metode pengukuran arus primernya.

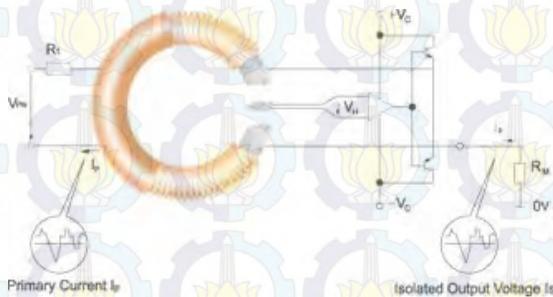


Gambar 2.13 Sensor tegangan LV 25-P[11]

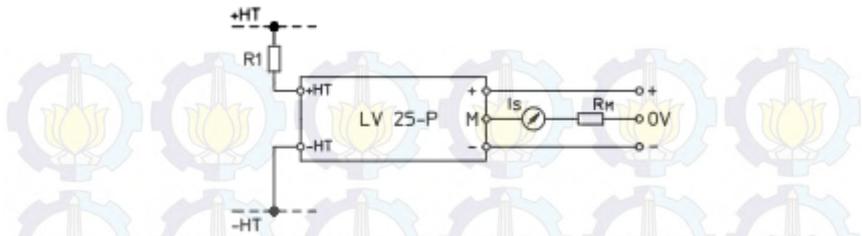
Perbedaan utama dari sensor arus ialah adanya tambahan kumparan primer internal, dengan jumlah lilitan yang besar agar terpenuhinya *ampere*turns* yang dibutuhkan, sehingga mampu mengukur arus primer yang kecil. Arus primer, I_P di dapat dari tegangan yang akan diukur, dengan dilewatkan pada resistor secara seri (R_1) terlebih dahulu. Pada gambar 2.14 terlihat desain dari *closed loop hall effect voltage transducer*[9].

2.3.1 Rangkaian Pengukuran Dasar

Sensor LV 25-P memiliki 5 pin, 5 pin tersebut yaitu pin +HT, -HT, +, M, dan -. Gambar 2.15 menunjukkan rangkaian pengukuran dasar sensor LV 25-P. Sesuai dengan *datasheet*, pemasangan R_M bernilai 30 – 100 Ohm pada arus primer I_P +/- 14mA *max*, dengan sumber tegangan +VC = 12V dan -VC= -12V. Akurasi dari sensor ini bergantung dari I_P . Sebisa mungkin R_1 dihitung terhadap tegangan nominal yang akan diukur agar mendapatkan I_P bernilai 10 mA (sesuai *datasheet*).



Gambar 2.14 *Closed loop hall effect voltage transducer*[12]



Gambar 2.15 Rangkaian pengukuran dasar LV 25-P[9]

2.4 Mikrokontroler

Mikrokontroler sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang kecil serta dapat diproduksi secara massal (dengan jumlah banyak) membuat harga menjadi murah dibandingkan mikroprosesor. Tidak seperti sistem komputer, yang mampu menangani berbagai macam program aplikasi, mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja.

Secara teknis, hanya ada 2 macam mikrokontroler. Pembagian ini didasarkan pada kompleksitas instruksi-instruksi yang dapat diterapkan pada mikrokontroler tersebut. Pembagian itu yaitu *RISC* dan *CISC*. *RISC* merupakan kependekan dari *Reduced Instruction Set Computer*. Instruksi yang dimiliki terbatas, tetapi memiliki fasilitas yang lebih banyak. Sebaliknya, *CISC* kependekan dari *Complex Instruction Set Computer*. Instruksi bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya.

Adapun kelebihan dari mikrokontroler adalah sebagai berikut:

1. Penggerak pada mikrokontroler menggunakan bahasa pemrograman *assembly* dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem.
2. Mikrokontroler tersusun dalam satu chip dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan control sistem sehingga mikrokontroler dapat di katakana computer mini.
3. Sistem *running* bersifat berdiri sendiri tanpa tergantung dengan computer.
4. Tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.

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

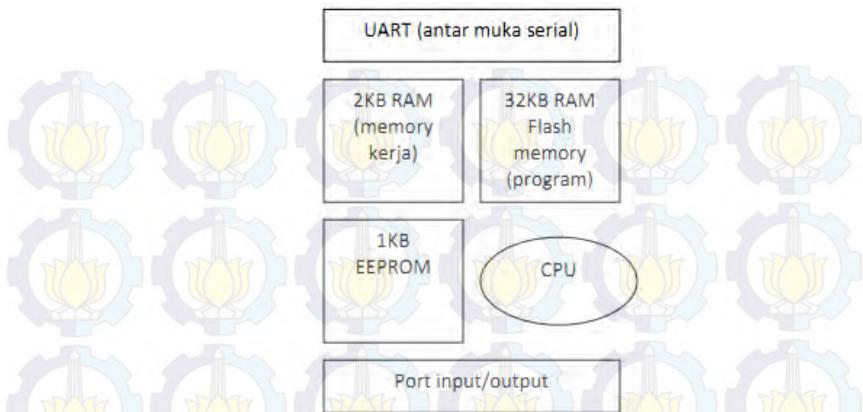
Mikrokontroler AVR menjadi chip utama pada alat pengembangan prototype Arduino. Arduino dikatakan sebagai sebuah platform dari *physical computing* yang bersifat *open source*. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari hardware, bahasa pemrograman dan *Integrated Development Environment (IDE)* yang canggih. IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam memory microcontroller. Ada banyak proyek dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena ia menjadi pilihan dan acuan bagi banyak praktisi.

Arduino dikembangkan oleh sebuah tim yang beranggotakan orang-orang dari berbagai belahan dunia. Anggota inti dari tim ini adalah:

- Massimo Banzi Milano, Italy
- David Cuartielles Malmoe, Sweden
- Tom Igoe New York, US
- Gianluca Martino Torino, Italy
- David A. Mellis Boston, MA, USA

Komponen utama di dalam papan Arduino adalah sebuah mikrokontroler 8 bit dengan merk ATmega yang dibuat oleh perusahaan Atmel Corporation. Berbagai papan Arduino menggunakan tipe ATmega yang berbeda-beda tergantung dari spesifikasinya, sebagai contoh Arduino Uno menggunakan ATmega328 sedangkan Arduino Mega 2560 yang lebih canggih menggunakan ATmega2560[13].

Arduino Uno adalah *board* mikrokontroler berdasarkan ATmega328. Pada gambar 2.16 ditunjukkan diagram blok sederhana dari mikrokontroler Atmega328. Dalam bahasa Itali "Uno" berarti satu,

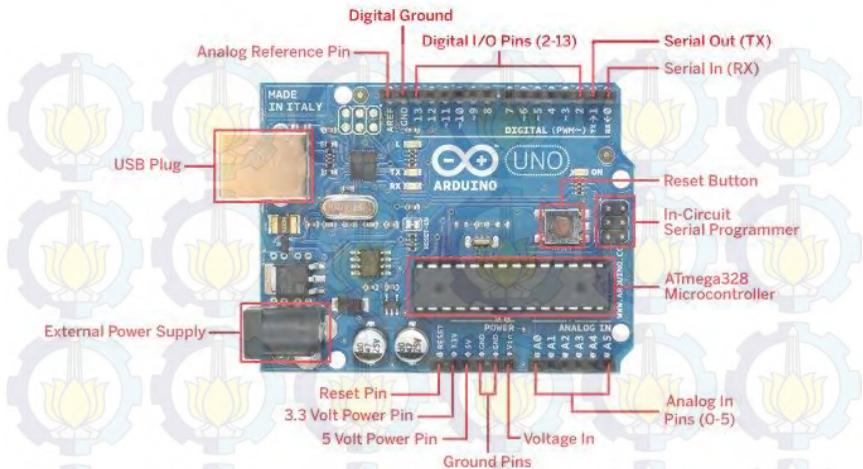


Gambar 2.16 Diagram blok Atmega328[13]

maka peluncuran Arduino 1.0 diberi nama Uno. Arduino ini berisi semua yang diperlukan untuk mendukung mikrokontroler, untuk mengaktifkannya cukup menghubungkannya ke komputer dengan kabel USB atau dengan adaptor AC 220v – DC 9V. Untuk pin I/O dari Arduino Uno, ditunjukkan pada gambar 2.18.

Blok-blok pada gambar 2.16 dijelaskan sebagai berikut[13]:

- Universal Asynchronous Receiver/Transmitter (UART) adalah antar muka yang digunakan untuk komunikasi serial seperti pada RS-232, RS 422 dan RS-485.
- 2KB RAM pada memory kerja bersifat volatile (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program.
- 32KB RAM flash memory bersifat non-volatile, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, flash memory juga menyimpan bootloader. Bootloader adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah bootloader selesai dijalankan, berikutnya program di dalam RAM akan dieksekusi.
- 1KB EEPROM bersifat non-volatile, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan.
- Central Processing Unit (CPU), bagian dari microcontroller untuk menjalankan setiap instruksi dari program.
- Port input/output, pin-pin untuk menerima data (input) digital atau analog, dan mengeluarkan data (output) digital atau analog.

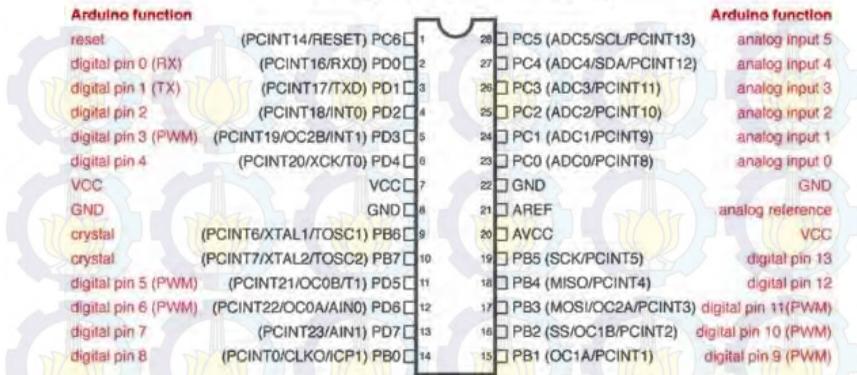


Gambar 2.17 Arduino Uno[14]

Spesifikasi dari Arduino Uno :

Mikrokontroler :	ATmega328
Operating Voltage :	5V
Input Voltage (disarankan) :	7-12V
Input Voltage (batas) :	6-20V
Digital I/O :	14 Pin (dimana 6 pin memberikan Output PWM)
Analog Input :	6 Pin
DC Current per I/O :	40 mA
DC Current for 3.3V :	50 mA
Flash Memory :	32 KB (ATmega328); 0,5 KB digunakan untuk Bootloader
SRAM :	2 KB (ATmega328)
EEPROM :	1 KB (ATmega328)
Clock Speed :	16 MHz

ATmega168/328 Pin Mapping



Gambar 2.18 Pin Mapping Arduino Uno[14]

2.5 Ethernet Shield

Ethernet *Shield* adalah modul yang dapat dipasangkan langsung di atas *board* Arduino UNO untuk menambahkan fungsi LAN / Ethernet sehingga memungkinkan arduino terhubung dengan internet. Dengan menggunakan modul komunikasi ini, Arduino mendukung protokol TCP/IP atau UDP.

Dengan memasang *shield* ini di atas *board* Arduino (ditumpuk) dan menyambungkan dengan kabel RJ45 dari suatu *network*, Arduino siap dikendalikan lewat intranet/internet. Tentunya dengan pemrograman yang telah dilakukan di IDE Arduino dan pemrograman ini terdiri dari 2



Gambar 2.19 Ethernet *Shield*[15]

jenis bahasa yaitu bahasa C untuk proses yang dilakukan mikrokontroler berkaitan dengan input/output dan bahasa HTML untuk membuat halaman web yang akan diakses.

Tegangan supply dari *shield* ini 5v DC yang didapat langsung dari board arduino. Spesifikasi *Ethernet Controllernya* yaitu chip Wiznet W5100 dengan internal buffer 16 Kb, kecepatan koneksi 10 / 100Mb (Fast-Ethernet). *Shield* ini terhubung dengan Arduino melalui port SPI.

Sebagai tambahan sudah tersedia slot micro-SD card yang dapat digunakan untuk menyimpan file yang akan disajikan ke jaringan. MicroSD card reader ini dapat diakses melalui SD Library (juga sudah tersedia di IDE Arduino).

2.6 Web Server Apache

Web server merupakan software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman web yang umumnya berbentuk dokumen HTML. Beberapa macam Web Server diantaranya:

1. Apache
2. Microsoft windows Server 2003 Internet Information Services (IIS)
3. Lighttpd
4. Sun Java System Web Server
5. Xitami Web Server
6. Zeus Web Server
7. Nginx

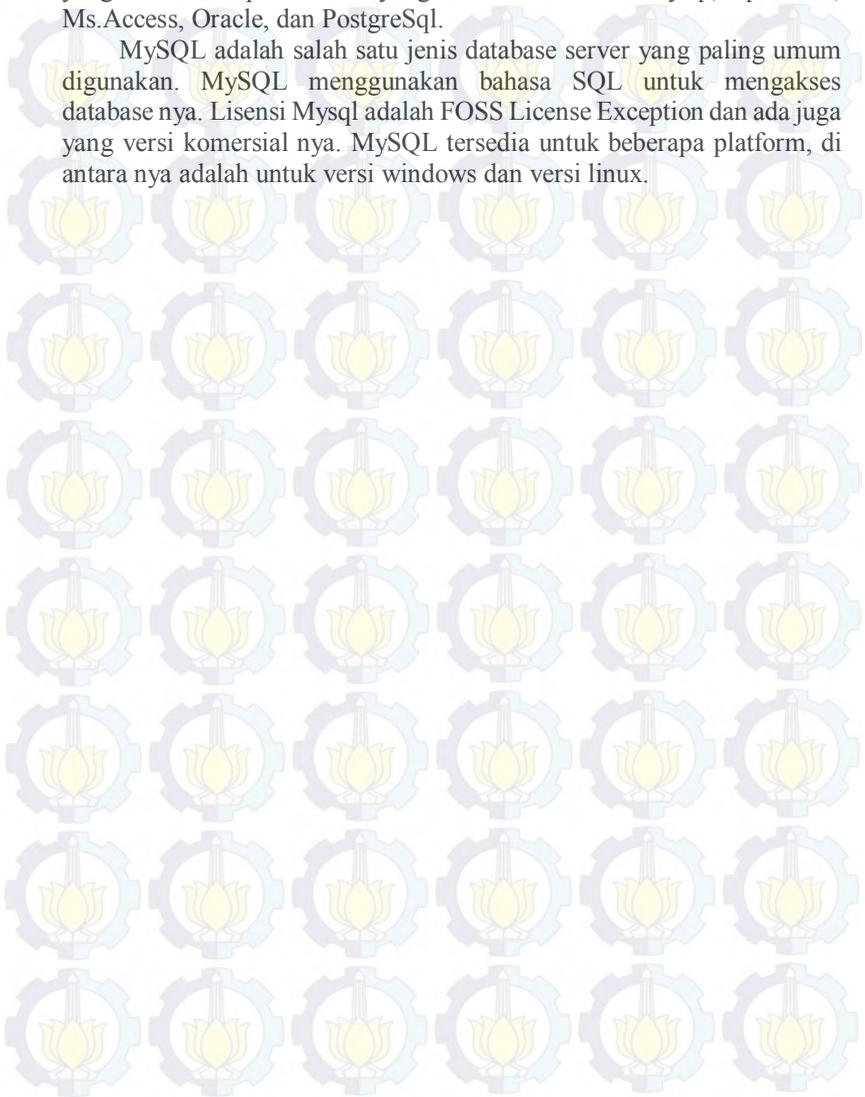
Apache merupakan web server yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya Apache mengeluarkan programnya yang dapat dijalankan di Windows NT. Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunaanya.

2.7 Database MySql

Database adalah sebuah system yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Database terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. Database digital di manage menggunakan Database Management System (DBMS), yang menyimpan isi database,

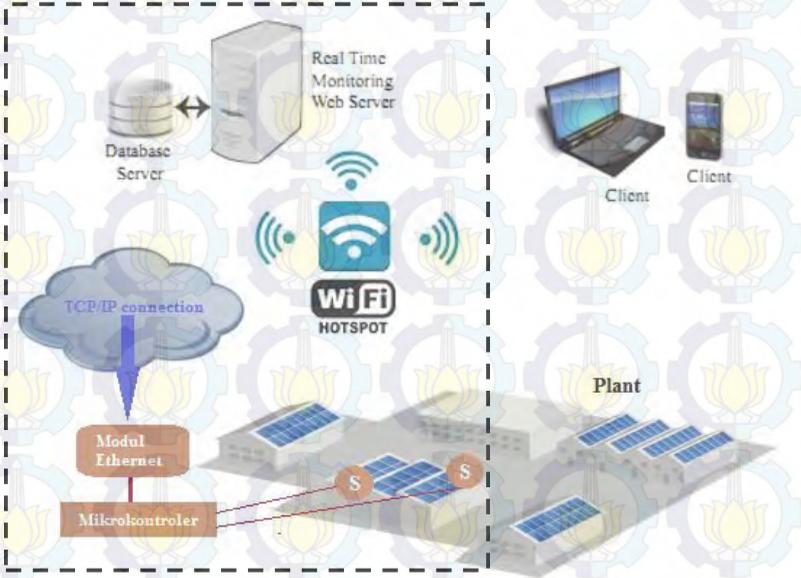
mengizinkan pembuatan dan maintenance data dan pencarian dan akses yang lain. Beberapa Database yang ada saat ini adalah : Mysql, Sql Server, Ms.Access, Oracle, dan PostgreSql.

MySQL adalah salah satu jenis database server yang paling umum digunakan. MySQL menggunakan bahasa SQL untuk mengakses database nya. Lisensi Mysql adalah FOSS License Exception dan ada juga yang versi komersial nya. MySQL tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux.



BAB III PERANCANGAN SISTEM

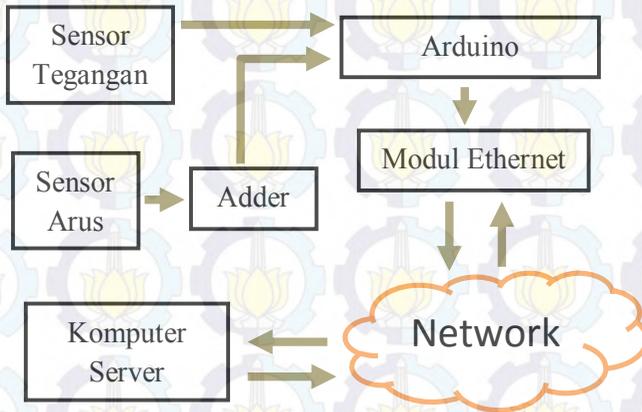
Pada bab ini akan dijelaskan terkait perancangan sistem secara keseluruhan, hal ini mencakup perancangan *hardware* dan perancangan *software*. Sensor arus dan tegangan diletakkan pada *plant* atau objek ukur, lalu sistem yang *menghandle* pembacaan data-data sensor arus dan tegangan ialah mikrokontroler Arduino yang berbasis Atmega 328 melalui ADCnya. Arduino dipasang modul Ethernet (*ethernet shield*) untuk dapat berkomunikasi serta mengirim data via protokol TCP/IP ke sebuah Web Server setiap 10 detik di sebuah laptop. Di laptop itu juga database MySQL berada untuk menyimpan 1000 data terakhir dari kiriman data Arduino. Client yang ingin memonitor nilai dari sensor – sensor harus terkoneksi dengan jaringan yang sama, lalu mengakses alamat IP statis dari Web Server yang telah disebutkan. Ilustrasi cara kerja sistem ditunjukkan pada gambar 3.1. Di dalam kotak putus-putus pada gambar 3.1 ialah sistem yang dibuat dalam tugas akhir ini.



Gambar 3.1 Ilustrasi cara kerja sistem

3.1 Diagram Blok Sistem

Cara kerja dari sistem secara umum digambarkan pada blok diagram pada gambar 3.2.



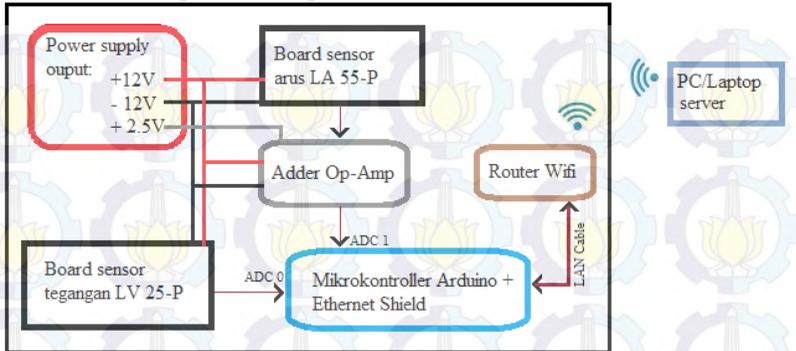
Gambar 3.2 Diagram blok sistem monitoring *charging station* berbasis web server

Berikut ini merupakan cara kerja keseluruhan sistem :

Sensor tegangan dan sensor arus diberi sumber tegangan $\pm 12\text{ V}$. Untuk sensor arus, outputnya diinputkan ke rangkaian adder dari Op-Amp agar level tegangan outputnya berada di nilai 2.5 V ketika tidak ada arus yang diukur. Hal ini dilakukan karena arus yang akan diukur memiliki kemungkinan arah yang berubah-ubah (sesuai keadaan objek ukur), sehingga ketika yang diukur arus negatif, maka outputnya akan berkisar antara $0 - 2,5\text{V}$, sedangkan jika arus positif yang diukur maka outputnya antara $2.5 - 5\text{V}$.

Data analog dari hasil pembacaan sensor tegangan dan arus yaitu berupa tegangan output antara $0 - 5\text{v}$ akan masuk ke pin ADC Arduino. Data analog tersebut diolah dan dirubah nilainya sesuai dengan hasil pengujian dan di linearisasi, kemudian setelah didapat nilai yang mencerminkan nilai input arus dan tegangan, akan dikirim ke komputer server yang memiliki alamat IP statik pada jaringan yang sama melalui modul ethernet. Hasil penghitungan tersebut masuk ke Database server setiap sepuluh detik dan akan ditampilkan di web secara realtime dan juga dalam bentuk tabel dan grafik 1000 data terakhir.

3.2 Perancangan Perangkat Keras (*Hardware*)

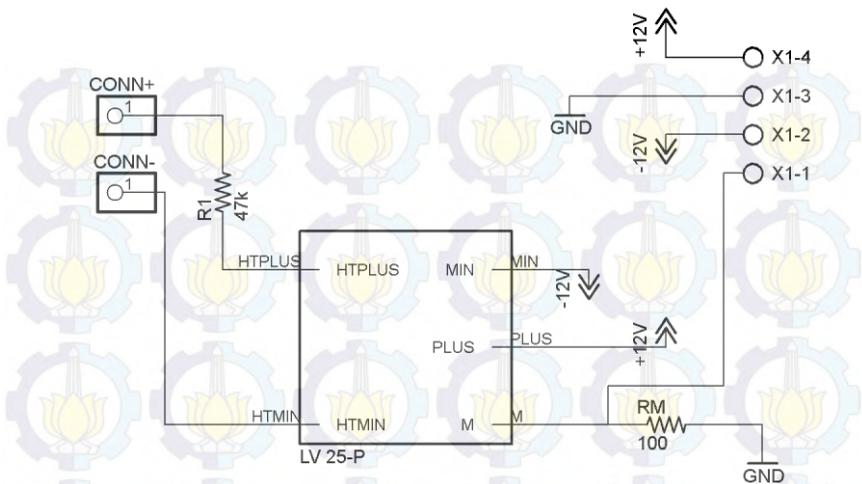


Gambar 3.3 Diagram blok *Hardware* sistem

Sebagaimana terlihat pada gambar 3.3, hal yang belum disinggung pada bab – bab sebelumnya ialah bagian power supply, router wifi dan adder Op-Amp. Blok power supply memiliki 3 buah output tegangan yaitu +12V, -12V, dan +2.5V (tentunya juga ada pin ground). Untuk +12V dan -12V diperlukan untuk memberi suplai tegangan catu ke rangkaian sensor arus LA 55-P dan sensor tegangan LV 25-p, juga ke rangkaian adder Op-Amp. Untuk +2.5V diperlukan bagi adder Op-Amp sebagai referensi adder. Router Wifi digunakan untuk membuat jaringan lokal agar ethernet *shield*, komputer server, dan client lain dapat berada pada suatu jaringan yang sama.

Output dari rangkaian sensor tegangan LV 25-P langsung dimasukkan ke ADC mikrokontroler karena nilainya tidak akan minus (dikondisikan pada objek ukur), berbeda dengan rangkaian sensor arus LA 55-P yang harus melalui rangkaian adder agar level tegangannya naik 2.5V. Hal ini sesuai dengan keadaan yang mungkin terjadi di lapangan bahwa arah arus dapat berubah (plus dan minus) sehingga dengan begitu akan aman untuk mikrokontroler.

Koneksi dari *ethernet shield* ke router wifi menggunakan kabel LAN karena ethernet *shield* hanya mendukung konektor fisik RJ45 agar dapat terhubung dalam suatu *network*. Sedangkan koneksi routern dengan PC/laptop server dimana database dan webserver berada adalah via wifi. Device lain yang ingin melihat nilai dari sensor arus dan tegangan dapat berupa komputer maupun *smartphone* via browser, yang terkoneksi pada router yang sama .

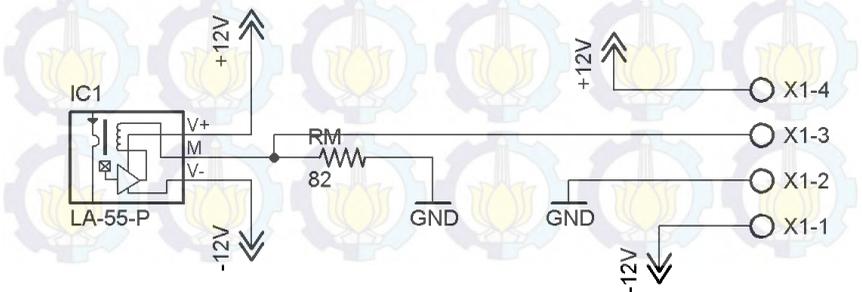


Gambar 3.5 Skematik rangkaian sensor tegangan LV 25-P

3.2.3 Rangkaian Sensor Arus LA 55-P

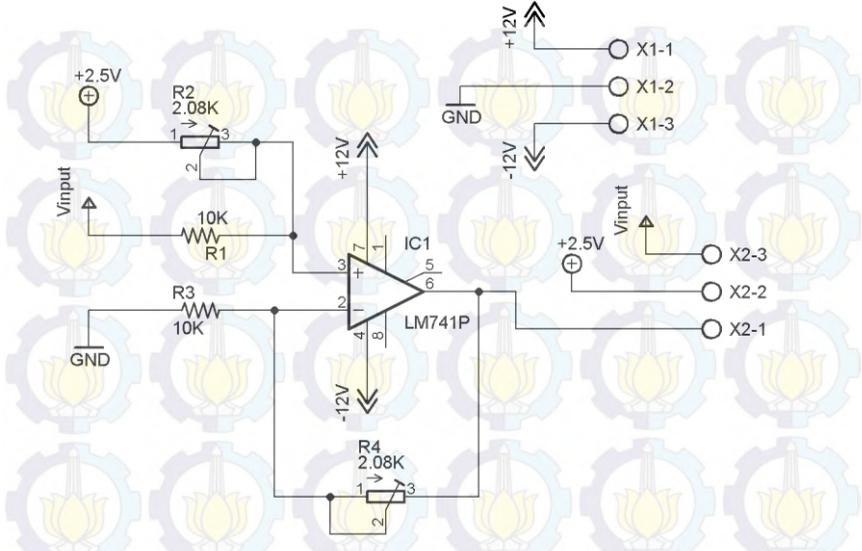
Sesuai dengan datasheet dari LA 55-P, maka R_M dipilih berkisar antara 10 – 100 Ω , penulis memilih 82 Ω . Tidak ada R_1 seri pada rangkaian ini jika dibandingkan dengan rangkaian sensor tegangan karena arus langsung dilewatkan melalui lubang yang tersedia pada sensor. Tegangan yang ada pada kabel yang dilewatkan ke dalam lubang sensor ini dibatasi sampai 500V.

Skematik dari rangkaian sensor arus ini ditunjukkan pada gambar 3.6. Pin konektor X1-3 merupakan pin output dari rangkaian ini (berupa tegangan), Sedangkan pin X1-4, X1-2, dan X1-1 merupakan pin catu daya yaitu +12V, GND, dan -12V yang didapat dari bagian *power supply*.



Gambar 3.6 Skematik rangkaian sensor arus LA 55-P

3.2.4 Rangkaian Adder (*Summing Amplifier*)



Gambar 3.7 Skematik rangkaian *summing amplifier*

Rangkaian *summing amplifier* diatas dapat dirumuskan hubungan input dan outputnya seperti berikut :

$$V_o = \left(V_1 \cdot \frac{R_2}{R_1 + R_2} + V_2 \cdot \frac{R_1}{R_1 + R_2} \right) \cdot \left(1 + \frac{R_4}{R_3} \right)$$

Dimana :

$$V_1 = V_{input}$$

$$V_2 = 2.5V$$

$$R_1 = R_3 = 10K$$

$$R_2 = R_4 = 2.08K$$

$$V_o = V_{output}$$

Dengan begitu jika V_{input} bernilai +12V, maka V_{output} akan bernilai 4.996V sedangkan jika V_{input} bernilai 0V, maka V_{output} akan bernilai 2.5V, dan jika V_{input} bernilai -12V, maka V_{output} akan bernilai 0.004V. Dengan demikian, pin mikrokontroler pada bagian ADC 1 akan dapat dengan aman mengukur tegangan output dari rangkaian sensor arus

LA 55-P, karena telah dikonversi level tegangannya terlebih dahulu oleh *summing amplifier* sesuai perhitungan diatas.

3.2.5 *Interfacing antara mikrokontroler degangan ethernet shield*

Sebagaimana dijelaskan pada BAB II, Ethernet *shield* ditumpukkan diatas arduino agar arduino dapat berkomunikasi dengan *device* lain melalui jaringan (*network*), bahkan dapat membuat arduino dapat terkoneksi dengan internet. Dengan menggunakan modul ethernet ini, arduino dapat berkomunikasi dengan perangkat lain dengan protokol TCP/IP atau UDP.

Ethernet *shield* ini bekerja berdasarkan chip ethernet Wiznet W5100, Arduino berkomunikasi dengan chip ini menggunakan BUS SPI atau dengan kata lain melewati pin ICSP header. Pin header ini ialah pin digital nomor 10,11,12,13 untuk arduino Uno. Jadi ketika Arduino bekerja dengan ethernet *shield* ini dan menjalankan komunikasi via *network*, maka pin digital yang telah disebutkan diatas tidak bisa digunakan untuk I/O.

Untuk menghubungkan arduino dengan etehernet *shield* ini dengan suatu jaringan, cukup menancapkan kabel network RJ45 yang dapat diambil dari router atau switch ke socket RJ45 *female* yang tersedia pada ethernet *shield*, lalu mengkonfigurasinya melalui program pada IDE arduino[16].



Gambar 3.8 Ethernet *shield* yang terhubung dengan mikrokontroler dan kabel LAN

Power dari Ethernet *shield* ini mengambil dari pin 5v dari arduino. Jadi selama arduino mendapat sumber daya yang cukup, maka ethernet *shield* dapat bekerja.

3.2.6 Wifi Router TP-Link TL-MR3020

Router yang digunakan dalam tugas akhir ini ialah router portable dengan disertai Wifi TL-MR3020 dari TP-Link. Dengan pertimbangan ukurannya yang relatif kecil, dengan dimensi 74x67x22 mm, dan relatif ringan. Powernya dapat berasal dari dari USB laptop karena inputnya membutuhkan tegangan 5v, tentu saja dapat juga dihidupkan dari adaptor yang memiliki output 5v.

Memiliki konektor USB untuk ditancapkan modem 3G USB dan memiliki konektor RJ45. Router ini memiliki 3 mode utama yaitu mode 3G, mode WISP, dan mode AP. Mode pertama yaitu mode 3G dapat dikatakan router ini dapat membuat jaringan wifi dengan paket data yang didapat dari modem 3G yang ditancapkan, tentu saja dapat juga dinikmati melalui kabel lan melalui konektor RJ45. Lalu selanjutnya mode WISP dan AP yang memungkinkan router ini menjadi repeater dan bridge dengan wifi lain dan juga dapat membuat koneksi network dari kabel LAN menjadi akses poin (dipancarkan melalui wifi). Spesifikasi lengkap dari router ini ialah[17] :

Hardware :

Interface : 1 10/100Mbps WAN/LAN Port, USB 2.0 Port for 3G/4G modem, a mini USB Port for power supply.

Button : Quick Setup Security Button, Reset Button, Mode Switch



Gambar 3.9 Portable 3G/4G Wireless Router TL-MR3020

External

Power Supply : 5VDC/1.0A

Antenna Type : Internal Antenna

Wireless Features :

Wireless

Standards : IEEE 802.11n, IEEE 802.11g, IEEE 802.11b

Frequency : 2.4-2.4835GHz

Transmit Power : <20dBm

Wireless Modes : 3G Router, Travel Router (AP), WISP Client Router

Wireless

Security : Support 64/128 bit WEP, WPA-PSK/WPA2 PSK, Wireless MAC Filtering

Software Features :

DHCP : Server, DHCP Client List, Address Reservation

Port

Forwarding : Virtual Server, Port Triggering, DMZ, UpnP

Security : Firewall, MAC filtering, Denial of Service (DoS)

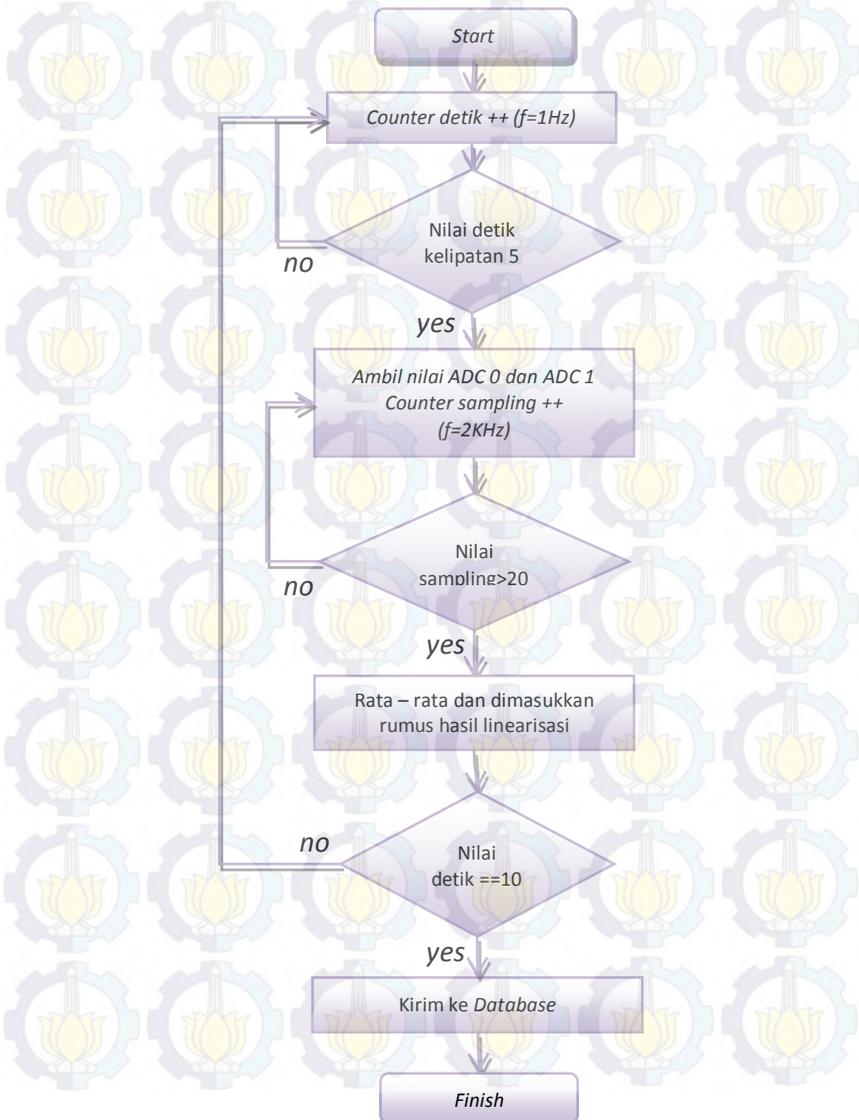
3.3 Perancangan Perangkat Lunak (*software*)

Perancangan perangkat lunak terdiri dari bagian program untuk sampling nilai ADC 0 dan ADC 1 serta bagian upload ke server. Pada saat pertama arduino *on*, maka inisiasi timer dilakukan.

Sesuai dengan *datasheet* Atmega 328, Uno memiliki tiga timer yang disebut timer0, Timer1, dan Timer2. Masing-masing timer memiliki counter yang bertambah pada setiap detik jam timer. Dalam tugas akhir ini akan dipakai mode timer CTC yaitu mode *clear timer on compare match*. Interrupt timer CTC dipicu ketika counter mencapai nilai tertentu yang disimpan dalam *compare math register*. Setelah timer counter mencapai nilai ini akan mereset (reset ke nol) pada clock timer berikutnya, selanjutnya akan terus menghitung sampai membandingkan nilai pada *compare math register* lagi. Dengan memilih nilai *compare math register* dan mengatur kecepatan di mana counter akan ter-*increment*, kita dapat mengontrol frekuensi interupsi timer.

Dalam *flowchart* pada gambar 3.10, dapat dilihat terdapat 2 interrupt yang digunakan, berfrekuensi 1Hz dan 2KHz. Hal ini dirancang agar

setiap detik ke 10 (dihandle oleh timer1 dengan frekuensi interrupt 1Hz), arduino mengirim data nilai tegangan dan arus yang terukur ke database

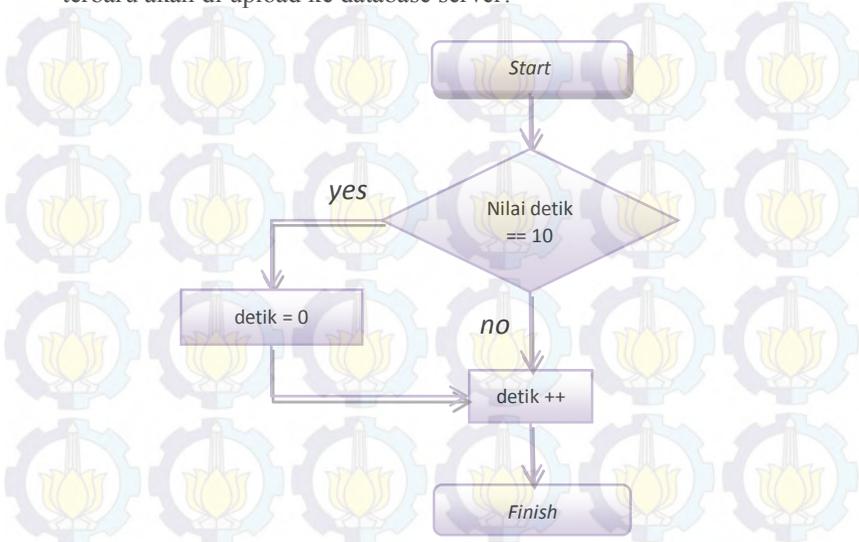


Gambar 3.10 Flow chart dalam sistem Arduino secara keseluruhan

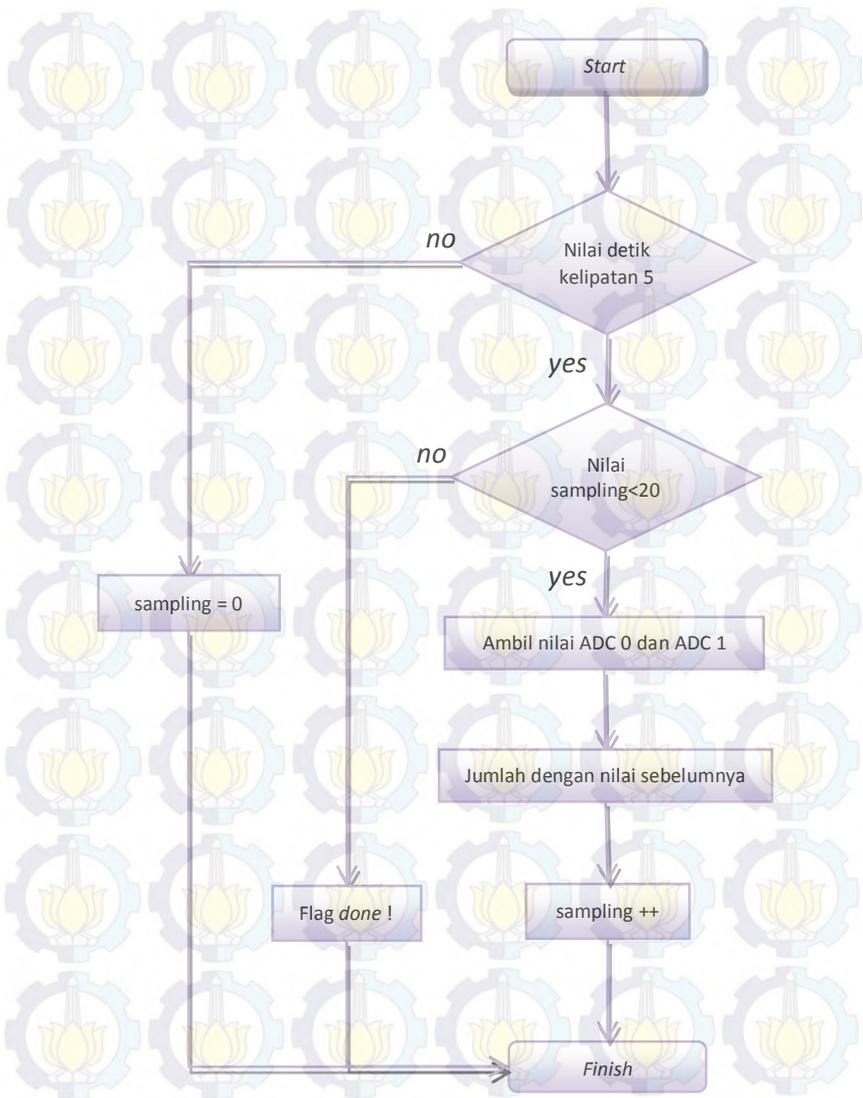
server. Data nilai tegangan dan arus tersebut didapat dari hasil olah nilai ADC0 dan ADC1 yang telah diolah dan dirumuskan dalam rumus linearisasi. Dan setiap 5 detik, maka arduino mensampling nilai ADC sebanyak 20 kali dengan frekuensi sampling 2KHz.

Jadi setiap interrupt pada timer 1 terjadi, yaitu setiap 1 detik (1Hz), maka akan ada sebuah variabel yang nilainya bertambah 1, bisa dikatakan hal ini representasi dari detik. Variabel tersebut akan terus naik nilainya dengan penambahan 1 hingga nilainya mencapai 10, kemudian akan dikembalikan lagi nilainya menjadi 0, dan *increment* lagi. Variabel detik tersebut merupakan variabel global, jadi nilainya dapat dicek di semua *scope* fungsi program. Hal ini ditujukan agar pada bagian *main loop* arduino, dapat terus mengecek apakah nilai detik sudah mencapai kelipatan 5, atau sudah mencapai nilai 10.

Jika nilai detik telah bernilai kelipatan 5, maka proses sampling data ADC0 dan ADC1 akan berlangsung, diambil sebanyak 20 kali penyamplingan dengan frekuensi 2KHz, kemudian di rata-rata dan di *display* kan di LCD 16x2 agar hasil nilai yang diolah dapat dilihat ditempat. Jika nilai detik telah mencapai nilai 10, maka hasil nilai yang terbaru akan di upload ke database server.

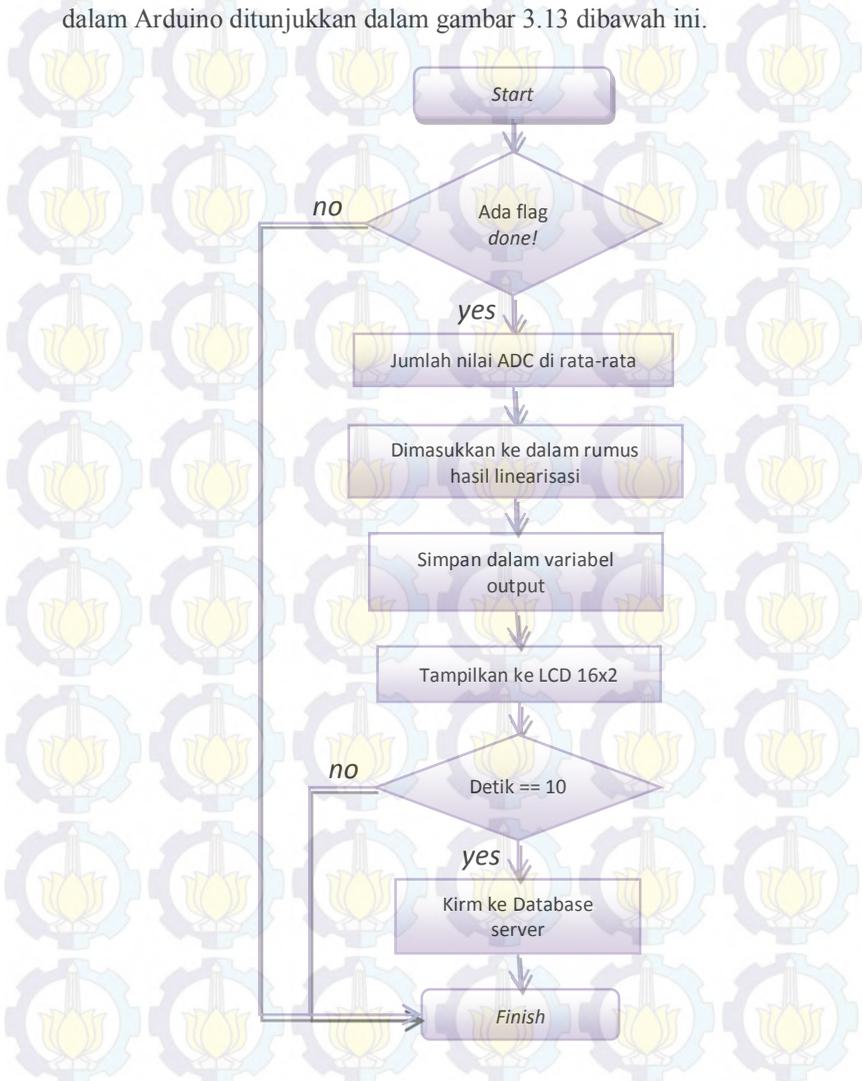


Gambar 3.11 Flow chart Interrupt Timer1, 1 Hz



Gambar 3.12 *Flow chart* Interrupt Timer2, 2KHz

Gambar 3.11 dan gambar 3.12 menunjukkan *flow chart* pada proses interrupt Timer1 dan Timer2. Sedangkan *flow chart* pada *loop* utama dalam Arduino ditunjukkan dalam gambar 3.13 dibawah ini.



Gambar 3.13 Flowchart *Loop* Arduino

Metode perhitungan nilai tegangan dan arus menggunakan rata-rata dari nilai sampling ADC. Nilai analog pada ADC diambil 20 kali, lalu di bagi 20 untuk mendapatkan rata-ratanya. Pada gambar 3.14 ditunjukkan bagian sampling ADC sebanyak 20 kali dengan frekuensi sampling 2 KHz. Untuk bagian nilai rata-rata dan pengkonversian nilai ADC ke nilai tegangan dari 0 – 5 V ditunjukkan pada gambar 3.15.

Pada gambar 3.15, tegangan referensi ADC menggunakan referensi supply dari mikrokontroler, yaitu 4.96V. Nilai tegangan tersebut di dapat dari pengukuran dengan multimeter pada pin catu daya mikrokontroler. Dengan resolusi ADC bernilai 10 bit, maka nilai *full scale* nya bernilai 1023.

Pada gambar 3.14 menunjukkan variabel jumlah0 merupakan penjumlahan dari nilai ADC0 yang sekarang dengan ADC0 yang lalu, begitu juga untuk variabel jumlah1. Pada gambar 3.16 menunjukkan kode program untuk mengirim data ke databse server dengan alamat IP statik 10.122.1.115.

```
ISR(TIMER2_COMPA_vect){ // 2 KHz -- 0.5 ms
    if (detik==5 && count<20){
        baca0=analogRead(0);
        baca1=analogRead(1);
        jumlah0=jumlah0+baca0;
        jumlah1=jumlah1+baca1;
        count=count+1;
    }
}
```

Gambar 3.14 Kode program bagian sampling ADC

```
tampung=(float)jumlah0/20.00;
tampung=tampung*4.96/1023.000;
```

Gambar 3.15 Kode program bagian rata-rata dan konversi ADC

```
if(detik==10){
  if (client.connect("10.122.1.115",80))
  {
    Serial.println("Sending to Server: ");
    client.println("POST /index.php/welcome/tambah HTTP/1.1");
    Serial.println("POST /index.php/welcome/tambah HTTP/1.1");
    client.println("Host: 10.122.1.76");
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.println("Connection: close");
    client.println("User-Agent: Arduino/1.0");
    client.print("Content-Length: ");
    client.println(temp.length());
    client.println();
    client.print(temp);
    client.println();
  }
  else
  {
    Serial.println("Cannot connect to Server");
  }
  client.stop();
}
```

Gambar 3.16 Kode program bagian upload nilai ke database server

BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan pengambilan data dan pengujian sistem. Pengambilan data ini berupa karakterisasi sensor tegangan dan sensor arus, juga menguji power supply dan rangkaian *summing amplifier*. Setelah dilakukan karakterisasi kemudian dilakukan pengujian sistem secara keseluruhan.

4.1 Pengujian Rangkaian Power supply

Pengujian ini bertujuan untuk mengetahui apakah *power supply* telah berfungsi, dengan memeriksa output tegangannya. Inputnya berupa tegangan 220V AC dari jala-jala. Penampakan dari rangkaian powers supply ditunjukkan pada gambar 4.1. Output dari power supply diuji dengan multimeter dan nilainya tegangannya tampak pada tabel 4.1.



Gambar 4.1 Rangkaian *power supply*



Gambar 4.2 Hasil pengujian tegangan output +12V *power supply*

Tabel 4.1 Hasil pengujian tegangan output *power supply*

Pin	Nilai Terukur (V)
12V	12.06
-12V	-11.84
2.5V	2.513

4.2 Pengujian Rangkaian Summing Amplifier

Pengujian rangkaian summing amplifier dengan cara memberi tegangan input dari 0V, +12v hingga -12V. Hasilnya dapat dilihat pada tabel 4.2. IC Op-Amp menggunakan lm741 dengan tegan sumbernya +-12V.



Gambar 4.3 Rangkaian *summing amplifier*

Hasil yang didapat, yaitu jika tegangan inputnya 0V, maka tegangan outputnya bernilai 2.559V, sedangkan pada saat tegangan inputnya dinaikkan hingga 12V, maka tegangan outputnya bernilai 5.02V. Dan jika tegangan output diturunkan hingga -12V, maka tegangan outputnya mencapai hampir mendekati 0V yaitu 10.9mV. Dengan hasil seperti ini, maka rangkaian suming aplifier ini telah dapatmengkoversi tegangan input dari rang -12V - +12V menjadi 0 – 5V DC, yang selanjutnya dapat diolah pada ADC mikro. Penampakan rangkaian summing amplifier ditunjukkan pada gambar 4.3. Untuk data pengujian rangkaian summing amplifier ini dapat dilihat pada tabel 4.2

Percobaan pengujian rangkaian summing amplifier ini menggunakan tegangan input dari *Laboratory DC Power Supply*, penampakannya pada gambar 4.4.



Gambar 4.4 *Laboratory DC Power Supply*

Tabel 4.2 Hasil pengujian rangkaian *summing amplifier*

Input (V)	Output (V)
0	2.559
3	3.172
6	3.79
9	4.41
12	5.02
-3	1.92
-6	1.282
-9	0.643
-12	0.01

4.3 Karakterisasi Sensor Tegangan LV 25-P



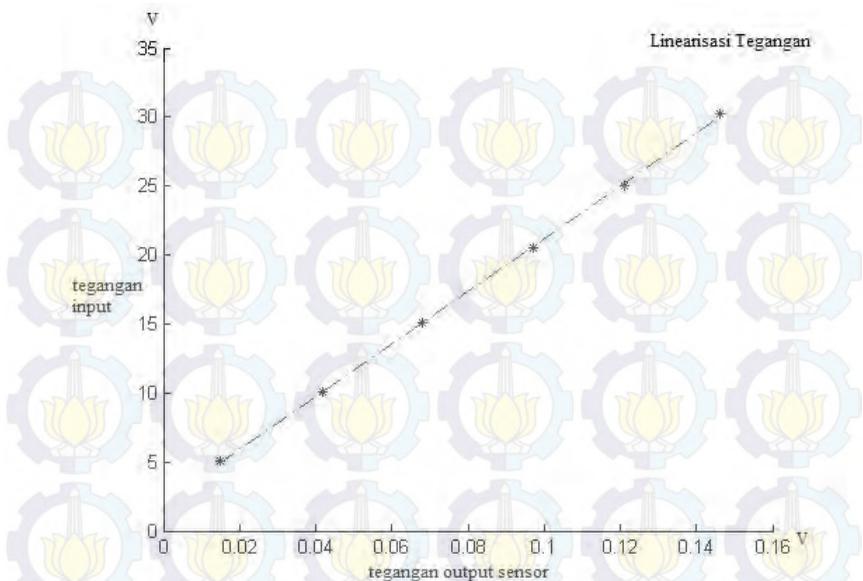
Gambar 4.5 Rangkaian sensor tegangan LV 25-P

Penampakan rangkaian sensor tegangan LV 25-P terlihat pada gambar 4.5. Pengujian dilakukan terhadap rangkaian sensor tegangan LV 25-P untuk mengetahui nilai tegangan output yang diberi tegangan input dari *Laboratory DC Power Supply*. Hasil pengujiannya ditunjukkan pada tabel 4.3.

Tabel 4.3 Data hasil pengujian rangkaian sensor tegangan LV 25-P

Tegangan Input (V)	Tegangan Output (V)
5.04	0.015
10.1	0.042
15.08	0.068
20.54	0.097
24.99	0.121
30.24	0.146

Kemudian dilakukan linierisasi untuk mendapatkan persamaan tegangan output terhadap tegangan input. Persamaan ini kemudian dimasukkan kedalam mikrokontroler untuk mendekati nilai tegangan input yang sesungguhnya. Grafik linierisasi ditunjukkan pada gambar 4.6.



Gambar 4.6 Grafik data linierisasi sensor tegangan LV 25-P

Dari proses linierisasi didapatkan bahwa, nilai gradien (m) = 191.1892 dan nilai konstanta (c) = 2.0831. Dari kedua nilai tersebut didapatkan persamaan untuk mendekati tegangan input berdasarkan tegangan output. Lalu dimasukkan ke dalam kode program mikrokontroler untuk mendekati nilai tegangan sebenarnya yang terukur.

Berdasarkan persamaan garis lurus secara umum :

$$y = mx + c \quad (3)$$

$$\text{Tegangan terukur} = 191.1892 * \text{tegangan output sensor} + 2.0831 \quad (4)$$

Mengacu pada gambar 3.15, maka persamaan (4) dimasukkan ke kode program mikrokontroler :

```
tampung=(191.1892*tampung)+2.0831;
```

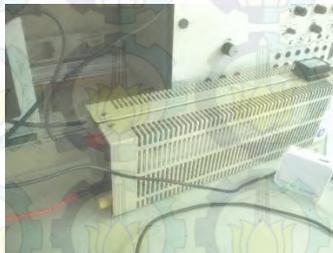
Gambar 4.7 Kode program perhitungan nilai tegangan yang terukur

4.4 Karakterisasi Sensor Arus LA 55-P

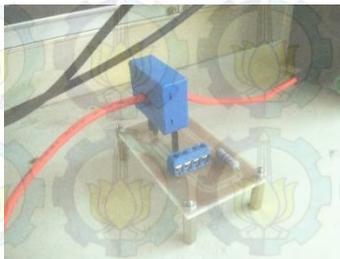
Penampakan rangkaian sensor tegangan LA 55-P terlihat pada gambar... Pengujian dilakukan terhadap rangkaian sensor tegangan LV 25-P untuk mengetahui nilai tegangan output yang diberi arus input dari *Laboratory DC Power Supply* yang diberi beban rheostat $10\ \Omega$ dengan arus maksimum yang dapat dilewati sebesar 5A. Hasil pengujiannya ditunjukkan pada tabel 4.4. Output tegangan dari rangkaian ini dimasukkan ke dalam rangkaian summing amplifier.



Gambar 4.8 Rangkaian sensor arus LA 55-P



Gambar 4.9 Rheostat $10\ \Omega$ yang digunakan



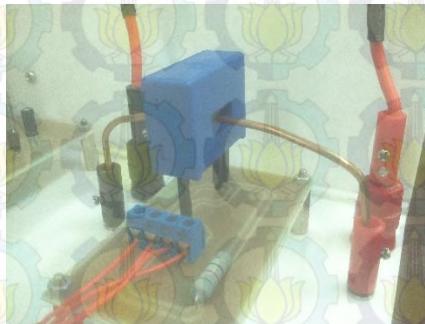
Gambar 4.10 Kabel dilewatkan di dalam lubang yang tersedia pada sensor LA 55-P

Dengan konfigurasi seperti pada Gambar 4.10, ternyata didapat tegangan output yang berubah – ubah bergantung pada posisi kabel. Posisi kabel di tengah dengan posisi kabel di pinggir lubang sensor LA 55-P memiliki output yang berbeda. Perbedaan ini dapat dilihat pada tabel 4.4.

Tabel 4.4 Data hasil pengujian rangkaian sensor arus LA 55-P dengan posisi kabel yang belum fix

Arus Input (A)	Kabel di Pinggir	Kabel di Tengah
	Tegangan Output (V)	Tegangan Output (V)
1	2.703	2.56
2	2.757	2.59
3	2.81	2.63

Dengan melihat hasil seperti ditunjukkan pada tabel 4.4, perbedaan tegangan output cukup besar, yaitu krang lebih 0.2 V. Kondisi yang seperti ini tentu tidak baik untuk pengukuran dilapangan, karena perubahan posisi kabel akan berpengaruh pada pembacaan. Berdasarkan hasil pada tabel 4.4, dibuatlah posisi konduktor yang fix untuk dilewati arus. Rancangan dari model ini dapat dilihat pada gambar 4.11. Kawat tembaga dilewatkan di tengah lubang sensor LA 55-P dengan posisi yang tetap, kawat tersebut berasal dari kabel tunggal berukuran 2 mm.

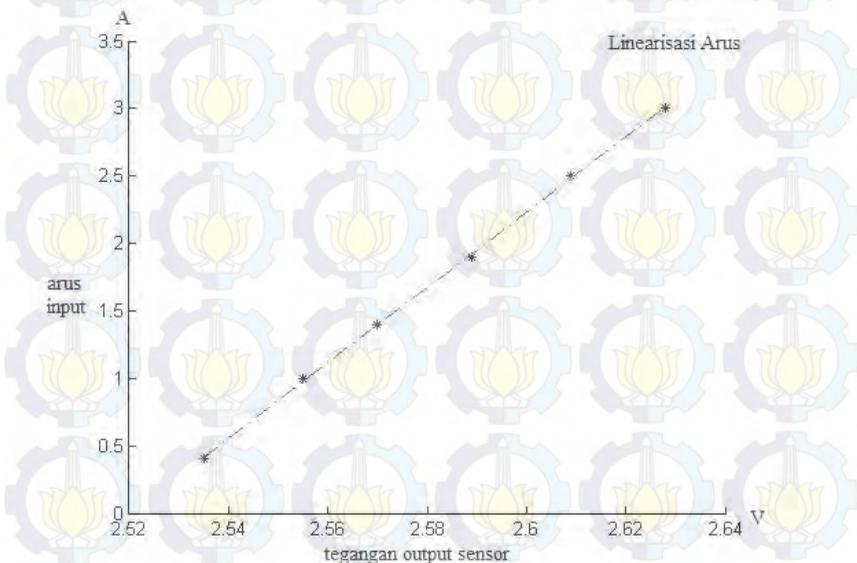


Gambar 4.11 Kawat tembaga dengan posisi yang tetap

Tabel 4.5 Data hasil pengujian rangkaian sensor arus LA 55-P fix

Arus Input (A)	Tegangan Output (V)
0.4	2.535
1	2.555
1.4	2.57
1.9	2.589
2.5	2.609
3	2.628

Kemudian dilakukan linierisasi untuk mendapatkan persamaan tegangan output terhadap arus input. Persamaan ini kemudian dimasukkan kedalam mikrokontroler untuk mendekati nilai arus input yang sesungguhnya. Grafik linierisasi ditunjukkan pada gambar 4.12.



Gambar 4.12 Grafik data linierisasi sensor arus LA 55-P

Dari proses linierisasi didapatkan bahwa, nilai gradien (m) = 27.8727 dan nilai konstanta (c) = -70.2394. Dari kedua nilai tersebut didapatkan persamaan untuk mendekati tegangan input berdasarkan tegangan output,

Berdasarkan persamaan garis lurus pada persamaan (3), didapat :

$$\text{Arus terukur} = 27.8727 * \text{tegangan output sensor} - 70.2394 \quad (5)$$

Mengacu pada gambar 3.15, maka persamaan (5) dimasukkan ke kode program mikrokontroller :

$$\text{tampung} = (27.8727 * \text{tampung}) - 70.2394;$$

Gambar 4.13 Kode program perhitungan nilai arus yang terukur

4.5 Pengujian Pembacaan Sensor Tegangan dan Sensor Arus

Pada bagian ini, sensor tegangan dan arus yang telah dilinearisasi akan diuji nilai outputnya terhadap nilai input yang sebenarnya, nilai input didapat dari sumber DC yang sama pada proses linearisasi. Hasilnya tampak pada tabel 4.6 untuk pembacaan tegangan tegangan, dan pada tabel 4.7 untuk hasil pembacaan arus. Serta grafiknya terlihat pada gambar 4.14 dan 4.15.

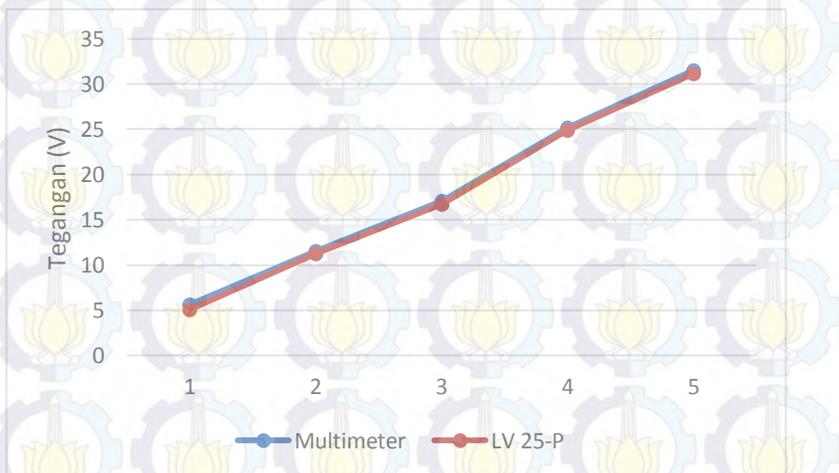
Tabel 4.6 Hasil pengujian nilai tegangan terukur setelah kalibrasi

Multimeter (V)	Hasil Pembacaan Mikrokontroller (V)	Error (%)
5.65	5.045	10.7
11.55	11.249	2.6
17.08	16.705	2.2
25.14	24.904	0.9
31.47	31.131	1.1

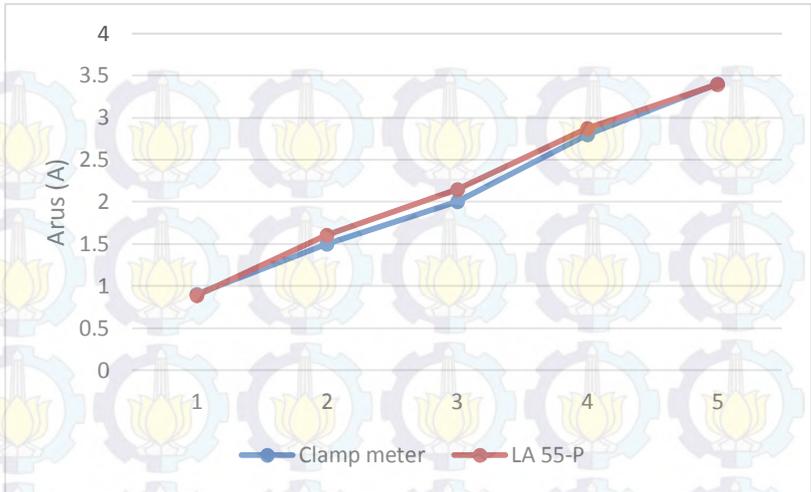
Tabel 4.7 Hasil pengujian nilai arus terukur setelah kalibrasi

Clamp meter (A)	Hasil Pembacaan Mikrokontroler (A)	Error (%)
0.9	0.885	1.7
1.5	1.607	7.1
2	2.149	7.5
2.8	2.872	2.6
3.4	3.397	0.1

Pengujian ini dilakukan dengan cara sumber tegangan *Laboratory DC Power Supply* diberi beban rheostat 10Ω . Dengan terminal sensor tegangan mengukur terminal tegangan input pada power supply, sedangkan sensor arus dilewatkan salah satu kabel yang menghubungkan rheostat dengan power supply. Hasil pengukuran ditampilkan pada LCD 16x2.



Gambar 4.14 Grafik pengujian sensor tegangan setelah kalibrasi



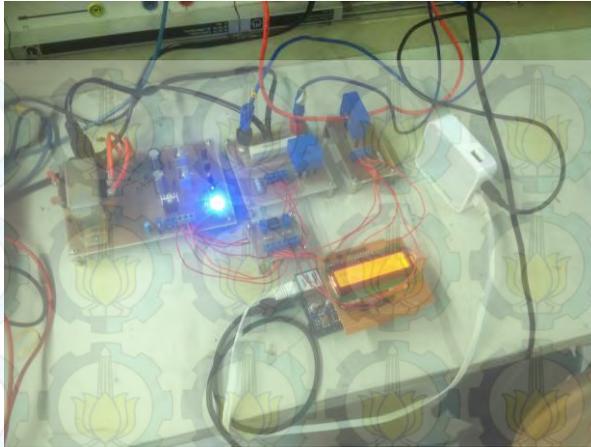
Gambar 4.15 Grafik pengujian sensor arus setelah kalibrasi



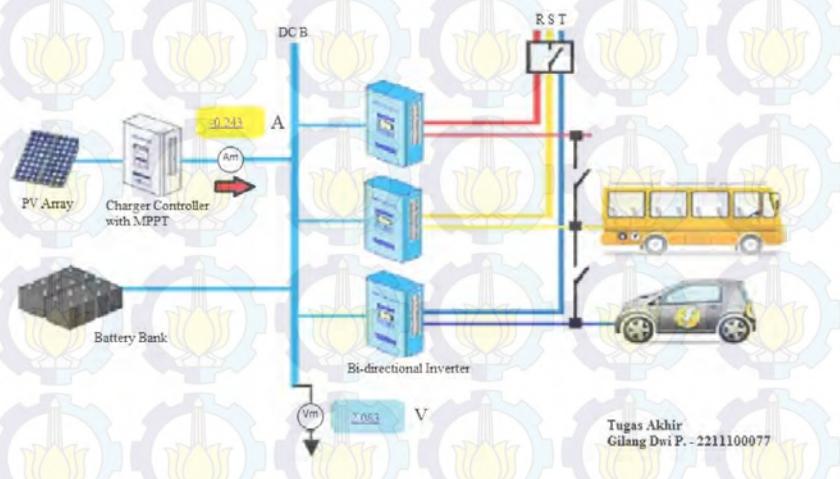
Gambar 4.16 Hasil pengukuran ditampilkan pada LCD 16x2

4.6 Pengujian Sistem Keseluruhan

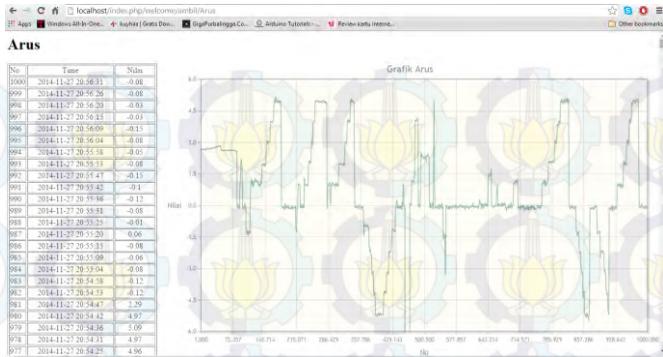
Pada pengujian sistem keseluruhan ini dilakukan dengan cara mengintegrasikan seluruh komponen sistem menjadi suatu sistem monitoring yang tampak seperti blok diagram pada **Gambar 3.3**, lengkap dengan router dan database server. Pengujian ini menguji pengiriman data yang telah diproses pada pengujian sebelumnya ke database server dan menguji data tersebut pada tampilan realtime monitoringnya.



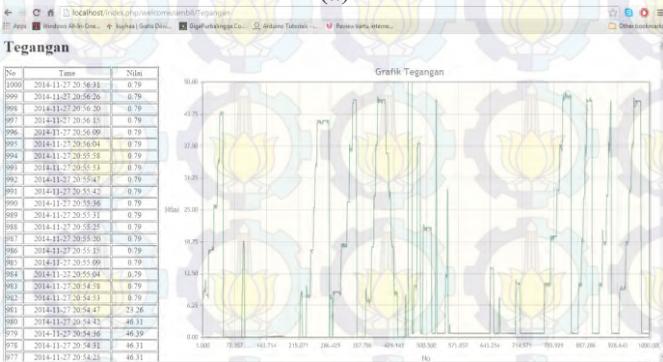
Gambar 4.17 Pengujian keseluruhan dengan router



Gambar 4.18 Tampilan real time monitoring pada webservice



(a)



(b)

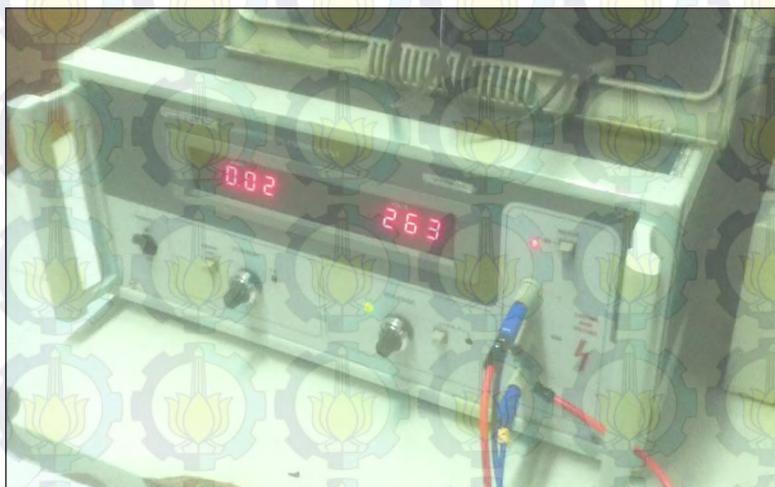
Gambar 4.19 Tampilan database pada webservice, nilai arus (a) nilai tegangan (b)

Pada database server, data yang disimpan hanya 1000 data terakhir, lengkap dengan tanggal dan jam ketika data tersebut masuk ke database, dan yang ditampilkan pada grafik adalah 1000 data tersebut. Sumbu x grafik tersebut merupakan indeks nomor (1-1000), dan sumbu y merupakan besar nilai tegangan / arus yang terukur. Dengan frekuensi upload data dari mikro ke database server setiap 10 detik, dengan kata lain 1000 data terakhir tersebut mewakili 2.8 jam terakhir. Memang untuk datalogger hal ini terbilang sedikit, tetapi parameter keberhasilan record data dari sistem ini telah terpenuhi.

4.7 Pengujian Pada Sumber DC lain

Pada pengujian ini dilakukan variasi tempat pengujian, yaitu untuk mengetahui kemampuan dari pembacaan tegangan dan arus yang relatif lebih tinggi. Untuk itu pengujian dilakukan terpisah terkait dengan lokasi dari sumber tegangan dan arus yang tinggi tersebut berbeda.

4.7.1 Pengujian Tegangan pada sumber DC di Laboratorium Konversi Energi Teknik Elektro ITS



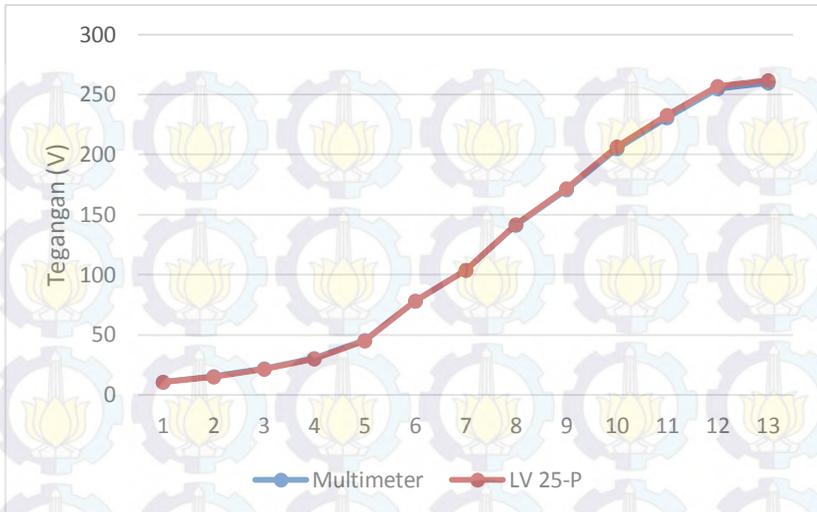
Gambar 4.20 Sumber DC yang mampu menyuplai tegangan hingga 263 V

Pada gambar 4.20 ditunjukkan sumber DC pada Lab Konversi energi yang dapat mencapai nilai 263 V DC, tetapi kemampuannya menyuplai arus sama dengan sumber DC yang sebelumnya, yaitu hanya sampai 3 A. Sumber ini kemudian digunakan untuk menguji sistem monitoring ini khususnya pada bagian pembacaan tegangan, dan hasilnya dapat dilihat pada tabel 4.8.

Tabel 4.8 Pengujian nilai tegangan terukur pada sumber tegangan DC lain

Multimeter (V)	Hasil Pembacaan Mikrokontroller (V)	Error (%)
10.67	10.387	2.65
15.61	14.952	4.22
21.91	21.458	2.1
30.45	30.021	1.4
45.3	45.182	0.26
78	78.095	0.12
103.9	104.051	0.15
141.4	141.918	0.37
170.8	171.766	0.57
205.1	206.761	0.81
231	232.854	0.8
254.8	257.048	0.88
259.8	261.683	0.72

Dikarenakan kemampuan dari sumber DC tersebut dalam menyuplai arus sama dengan sumber DC yang digunakan untuk linearisasi, maka untuk dalam pengujian kali ini yang diuji hanyalah nilai tegangan saja. Pada tabel 4.8 terlihat nilai maksimal dari sumber DC adalah 259.8 V, dan hasil pembacaan menunjukkan 261.683 V. Untuk grafik dari pengujian ini dapat dilihat pada gambar 4.21. Dari grafik tersebut terlihat garis pembacaan dari multimeter dan LV 25-P berhimpitan dan mempunyai error yang relatif kecil.



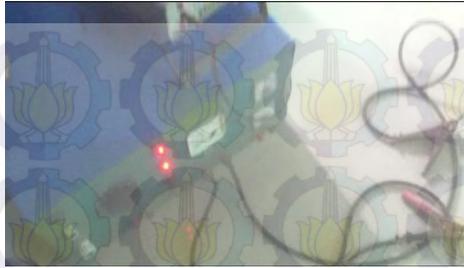
Gambar 4.21 Grafik pengujian sensor tegangan untuk tegangan yang relatif lebih tinggi

4.7.2 Pengujian Arus pada sumber DC di Laboratorium Tegangan Tinggi Teknik Elektro ITS

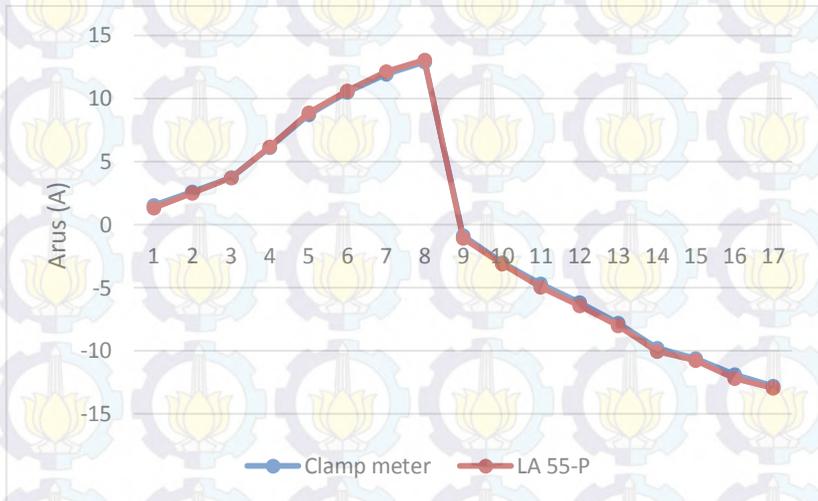
Pada pengujian di bagian ini, akan diuji nilai dari arus input yang relatif lebih besar daripada yang bisa disuplai oleh sumber DC sebelumnya. Kemampuan dari sumber DC kali ini mampu menyuplai arus hingga nilai kisaran 13 A. Penampakan dari *current injector* ini terlihat pada gambar 4.22. Pengujian pada bagian ini juga akan menguji arus egatif, yaitu dengan cara membalik terminal pemasangan pada sensor arus LA 55-P, untuk mengetahui nilai pembacaan arus pada arah yang berbeda. Hal ini terkait dengan kebutuhan di lapangan yang memungkinkan arah aliran arus listrik yang bisa berubah – ubah. Hasil data pengujian kali ini ditunjukkan pada tabel 4.9. Grafik dari pengujian ini dapat dilihat pada gambar 4.23. Pada gambar 4.21 terlihat hasil pembacaan Clamp meter dibandingkan dengan sensor arus LA 55-P berhimpitan, dengan warna garis biru menunjukkan nilai dari Clamp meter, dan garis merah menunjukkan nilai dari hasil pembacaan sensor arus LA 55-P.

Tabel 4.9 Pengujian nilai arus terukur pada sumber arus DC lain

Clamp meter (A)	Hasil Pembacaan Mikrokontroler (A)	Error (%)
1.5	1.304	13.0667
2.6	2.513	3.34615
3.7	3.696	0.10811
6.1	6.169	1.131148
8.7	8.872	1.977011
10.5	10.615	1.095238
11.9	12.115	1.806723
12.9	13.054	1.193798
-0.9	-1.061	17.88889
-3	-3.142	4.733333
-4.7	-4.96	5.531915
-6.2	-6.446	3.967742
-7.8	-7.994	2.487179
-9.8	-10.04	2.44898
-10.6	-10.79	1.792453
-11.9	-12.19	2.436975
-12.8	-12.94	1.09375



Gambar 4.22 Sumber DC yang mampu menyuplai arus hingga 13 A



Gambar 4.23 Grafik pengujian sensor arus untuk arus yang relatif lebih tinggi

BAB V

PENUTUP

5.1 Kesimpulan

Sistem monitoring yang telah disusun pada Tugas Akhir ini telah teruji dapat mengukur tegangan DC hingga 259.8 V dengan error 0.72% dan arus hingga 12.9 A dengan error 1.19%, sedangkan arus hingga -12.8 A dengan error 1.09%. Nilai tersebut didapat dari proses *analog to digital conversion* yang dilakukan oleh ADC mikrokontroler arduino uno dari keluaran sensor yang diambil sebanyak 20 kali kemudian di rata-rata, lalu dimasukkan ke dalam rumus orde satu hasil linearisasi sensor.

Selama database server terkoneksi satu jaringan dengan modul ethernet berada, maka pengiriman data dari ethernet modul ke server dapat berlangsung, melalui protokol TCP/IP. Jika database server tidak *on* maka pengecekan nilai yang terukur dapat dilihat pada LCD 16x2.

Data yang disimpan pada database server dibatasi hingga 1000 data terakhir. Dengan selang waktu pengiriman data ke database server setiap 10 detik. Tampilan pada web server berupa realtime nilai yang terukur pada saat itu juga, tabel 1000 data terakhir, dan grafik 1000 data terakhir.

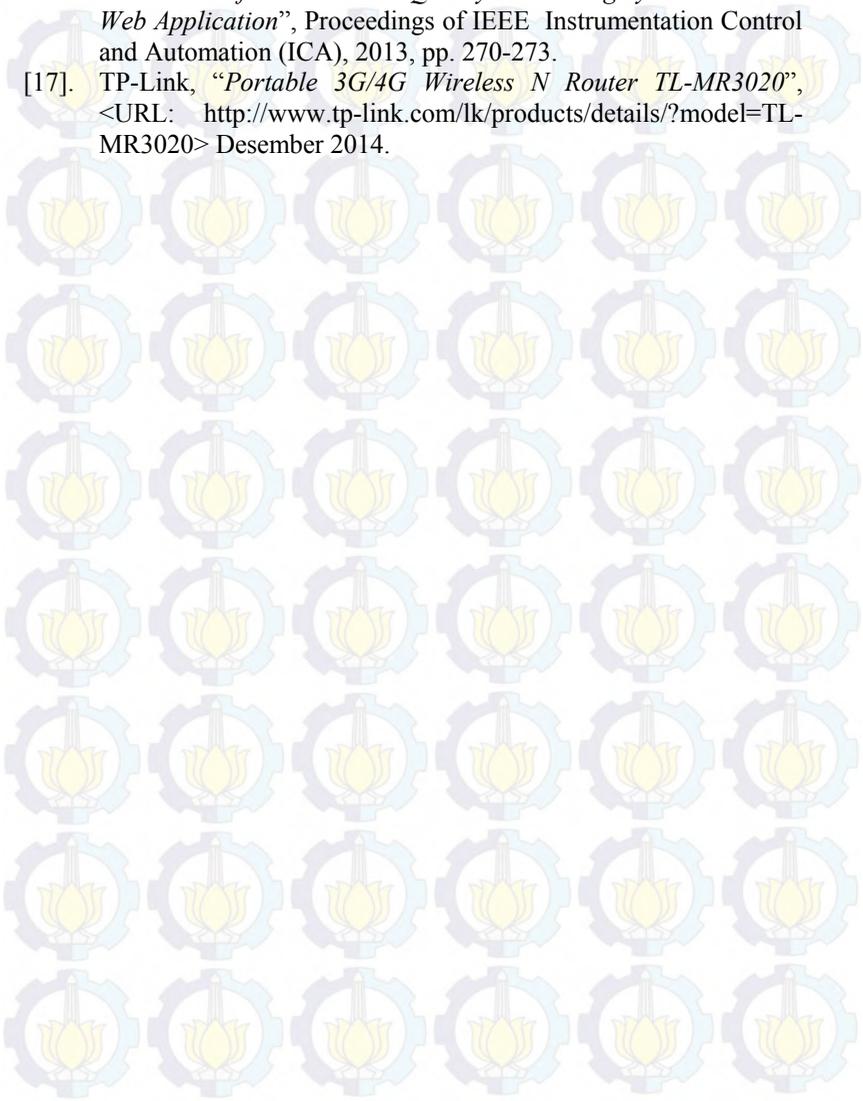
5.2 Saran

Berdasarkan pengujian dan percobaan yang dilakukan oleh penulis, maka pengembangan lebih lanjut dari sistem ini ialah sistem database server yang tidak harus berada dalam satu jaringan dengan modul ethernet, tetapi dalam bentuk domain. Jadi koneksi mikrokontroler arduino dengan database server merupakan koneksi internet. Dengan begitu ethernet modul hanya perlu koneksi internet yang stabil untuk dapat mengirimkan datanya ke domain, tanpa harus ada sebuah server berupa komputer yang harus selalu *on*.

DAFTAR PUSTAKA

- [1]. Hydro-Quebec, "*Electric Vehicle Charging Stations*", Technical Installation Guide, January 2012.
- [2]. _____, "*Electrical Vehicle (EV) Charging Station*", <URL: http://www.avinc.com/glossary/electric_vehicle_ev_charging_stations>, November 2014.
- [3]. _____, "*Universal EV Combined Charging System*", <URL: <http://www.torquenews.com/1075/electric-car-fast-charging-feud-between-sae-and-chademo-who-will-win>>, November 2014.
- [4]. Prabowo, Wiratno, Hantoro, Asfani, Rahmadiansah, "*Infrastruktur Charging Station*", Mobil Listrik Nasional ITS, 2014.
- [5]. _____, "*Tinker's Corner Workbench I: the WebDAQ/100*", <URL: <http://www.exisle.net/mb/index.php?/topic/59402-tinkers-corner-workbench-i-the-webdaq100/>> 27 June 2009.
- [6]. Datasheet CEC, "*webDAQ/100 Distributed Data Acquisition System*", Capital Equipment Corporation, Januari 2015.
- [7]. Glenn K. Young, P.E., "*Continuous Gas Monitoring*", 2015.
- [8]. Honeywell Inc, "*Hall Effect Sensing and Application*", Honeywell Manual, November 2014.
- [9]. LEM Components, "*Isolated Current and Voltage Transducers*", LEM Characteristic – Application - Calculation, November 2014.
- [10]. _____, "*Product of LA 55-P*", <URL: http://www.lem.com/hq/en/component/option,com_catalog/task,displaymodel/id,90.13.25.000.0>, November 2014.
- [11]. _____, "*Product of LV 25-P/SP5*", <URL: <http://www.wattbits.com/lv25-p-sp5.html>>, Desember 2014.
- [12]. _____, "*Voltage Transducer (Hall Close Loop)*" <URL: [http://www.passiontek.com.cn/english/productxl.asp?id=193&titlename=Voltage%20Transducers\(Hall%20Close%20Loop\)&lmtyp=30](http://www.passiontek.com.cn/english/productxl.asp?id=193&titlename=Voltage%20Transducers(Hall%20Close%20Loop)&lmtyp=30)>, Desember 2014.
- [13]. Feri Djuandi, "*Pengenalan Arduino*", Juli 2011.
- [14]. MrLndr, "*FAQ: The Arduino Anatomy*", <URL: <http://arduinoarts.com/2011/08/the-arduino-uno-anatomy/>>, 9 Agustus 2011.
- [15]. _____, "*Arduino Ethernet Shield Rev.3*", <URL: <http://www.cooking-hacks.com/arduino-ethernet-shield>>, Desember 2014.

- [16]. M. F. Ramadhani, B. Yulianto, Nugraha, “*Design and Fabrication of Real Time Air Quality Monitoring System Based on Web Application*”, Proceedings of IEEE Instrumentation Control and Automation (ICA), 2013, pp. 270-273.
- [17]. TP-Link, “*Portable 3G/4G Wireless N Router TL-MR3020*”, <URL: <http://www.tp-link.com/lk/products/details/?model=TL-MR3020>> Desember 2014.



LAMPIRAN

Program IDE Arduino

```
#include <Ethernet.h>
#include <SPI.h>
#include <LiquidCrystal.h>
#include <stdlib.h>
#include <string.h>
LiquidCrystal lcd(9, 8, 5, 6, 3, 2);

byte mac[] = {
  0x90, 0xA2, 0xDA, 0x0D, 0xF6, 0xFF };
byte ip[] = {
  10, 122, 1, 76};

EthernetClient client;
float tampung;
float sum=0;
int upload=0;
int baca0;
int baca1;
int jumlah0;
int jumlah1;
int count;
int maks0=0;
int minim0=1023;
int maks1=0;
```

```

int minim1=1023;
int selisih;
char buffer[10];
String temp= "data1=5&data2=7";
byte sampai=0;
byte sampling=0;
byte cukup=0;
byte go=0;
void setup()
{
cli();//stop interrupts
//set timer1 interrupt at 1Hz
TCCR1A = 0;// set entire TCCR1A register to 0
TCCR1B = 0;// same for TCCR1B
TCNT1 = 0;//initialize counter value to 0
// set compare match register for 1hz increments
OCR1A = 15624;// = (16*10^6) / (1*1024) - 1 (must be <65536)
// turn on CTC mode
TCCR1B |= (1 << WGM12);
// Set CS12 and CS10 bits for 1024 prescaler
TCCR1B |= (1 << CS12) | (1 << CS10);
// enable timer compare interrupt
TIMSK1 |= (1 << OCIE1A);

//set timer2 interrupt at 2kHz

```

```

TCCR2A = 0;// set entire TCCR2A register to 0
TCCR2B = 0;// same for TCCR2B
TCNT2 = 0;//initialize counter value to 0
// set compare match register for 2khz increments
OCR2A = 124;// = (16*10^6) / (2000*64) - 1 (must be <256)
// turn on CTC mode
TCCR2A |= (1 << WGM21);
// Set CS21 bit for 64 prescaler
TCCR2B |= (1 << CS22);
// enable timer compare interrupt
TIMSK2 |= (1 << OCIE2A);

sei();//allow interrupts

Ethernet.begin(mac, ip);
Serial.begin(9600);
Serial.println(Ethernet.localIP());
delay(2000);
Serial.println("connecting...");

lcd.begin(16, 2);
lcd.print("A0 (V)  A1 (A)");
}

ISR(TIMER1_COMPA_vect){ //change the 0 to 1 for timer1, 1HZ
//interrupt commands here
sampling=0;

```

```

if(upload==10){
  lcd.setCursor(7, 0);
  lcd.print(" ");
  upload=0;
  sampai=0;
  sampling=1;
  cukup=0;
}
if(upload==5){
  cukup=0;
  sampling=1;
}
upload=upload+1;
lcd.setCursor(7, 0);
lcd.print(upload);
}

ISR(TIMER2_COMPA_vect){//2 KHz -- 0.5 ms
  if (detik==5 && count<20){
    baca0=analogRead(0);
    baca1=analogRead(1);
    jumlah0=jumlah0+baca0;
    jumlah1=jumlah1+baca1;
    count=count+1;
  }
  else if (sampling==1 && count>19){

```

```
count=21;
}
else{
count=0;
}
}

void loop()
{
delay(50);
if(count==21 && cukup==0){
cukup=1;
lcd.setCursor(0, 1);
lcd.print("      ");
lcd.setCursor(0, 1);

tampung=(float)jumlah0/20.00;
tampung=tampung*4.96/1023.000;
tampung=(191.1892*tampung)+2.0831; //orde 1

lcd.print(tampung,3);
temp="data1=";
char tmp[7] = "";
dtostrf(tampung,1,3, &tmp[0]);
temp=temp+tmp;
```

```
lcd.setCursor(10, 1);
```

```
tampung=(float)jumlah1/20.00;
```

```
tampung=tampung*4.96/1023.000;
```

```
tampung=(27.8727*tampung)-70.2394; //orde1 lab KE dc asli
```

```
lcd.print(tampung,3);
```

```
temp=temp+"&data2=";
```

```
char tmp1[7] = "";
```

```
dtostrf(tampung,1,3, &tmp1[0]);
```

```
temp=temp+tmp1;
```

```
if(detik==10){
```

```
    if (client.connect("10.122.1.115",80))
```

```
    {
```

```
        Serial.println("Sending to Server: ");
```

```
        client.println("POST /index.php/welcome/tambah HTTP/1.1");
```

```
        Serial.println("POST /index.php/welcome/tambah HTTP/1.1");
```

```
        client.println("Host: 10.122.1.76");
```

```
        client.println("Content-Type: application/x-www-form-  
urlencoded");
```

```
        client.println("Connection: close");
```

```
        client.println("User-Agent: Arduino/1.0");
```

```
        client.print("Content-Length: ");
```

```
        client.println(temp.length());
```

```
        client.println();
```

```
        client.print(temp);
```

```
client.println();
}
else
{
  Serial.println("Cannot connect to Server");
}
client.stop();
}
Serial.println("calculate ");
Serial.println(temp);
jumlah0=0;
jumlah1=0;
maks0=0;
minim0=1023;
maks1=0;
minim1=1023;
go=0;
if (upload==10)go=1;
}
}
```



Halaman ini sengaja dikosongkan

Program style.css

```
body {  
    background: url('skema7.jpg');  
    background-repeat: no-repeat;  
    background-size: 1200px;
```

```
}  
#dua {  
    position: relative;  
    /*top: 205px;  
    left: 210px;*/  
    top: 110px;  
    left: 400px;  
}
```

```
#satu {  
    position: relative;  
    /*top: 345px;  
    left: 200px;*/  
    top: 492px;  
    left: 575px;  
}
```



Halaman ini sengaja dikosongkan

Program welcome.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
```

```
class Welcome extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *      http://example.com/index.php/welcome
     *    - or -
     *      http://example.com/index.php/welcome/index
     *    - or -
     * Since this controller is set as the default controller in
     * config/routes.php, it's displayed at http://example.com/
     *
     * So any other public methods not prefixed with an underscore
    will
     * map to /index.php/welcome/<method_name>
     * @see http://codeigniter.com/user_guide/general/urls.html
    */
    public function __construct()
    {
        parent::__construct();
        $this->load->model("data_model");
        //$this->load->library('session');
    }

    public function tambah()
    {
        $dpost = $this->input->post();
        $mysqltime = date ("Y-m-d H:i:s");
        $stampung = array("0" => array("nilai" =>
        $dpost['data1'], "time" => $mysqltime), "1" => array("nilai" =>
        $dpost['data2'], "time" => $mysqltime));
        $hasil = $this->data_model->tambah_data($stampung);
        $this->data_model->cek_data();
    }
}
```

```
public function index()
{
    $stampil = $this->data_model->tampil_data();
    $this->load->view('tampil',$stampil);
}

public function ambil($id)
{
    $stampil["data"] = $this->data_model-
->ambil_data($id);
    $stampil["judul"] = $id;
    $this->load->view('sub',$stampil);
}

/* End of file welcome.php */
/* Location: ./application/controllers/welcome.php */
```

Program data_model.php

```
<?php
class Data_model extends CI_Model {
    public function __construct()
    {
        $this->load->database();
    }
    public function tambah_data($data)
    {
        $this->db->insert('titik1', $data[0]);
        $this->db->insert('titik2', $data[1]);
    }
    public function tampil_data()
    {
        $query1 = "SELECT nilai from titik1 ORDER BY
time DESC LIMIT 1";
        $query2 = "SELECT nilai from titik2 ORDER BY
time DESC LIMIT 1";
        $res1 = $this->db->query($query1);
        $res2 = $this->db->query($query2);
        $res1 = $res1->result_array();
        $res2 = $res2->result_array();
        $hasil = array("nilai1" => $res1[0]["nilai"], "nilai2" =>
$res2[0]["nilai"]);
        return $hasil;
    }
    public function cek_data()
    {
        $jumlah1 = "select count(time) from titik1";
        $res_1 = $this->db->query($jumlah1);
        $res_1 = $res_1->result_array();
        $res_1 = $res_1[0]["count(time)"];

        $jumlah2 = "select count(time) from titik2";
        $res_2 = $this->db->query($jumlah2);
        $res_2 = $res_2->result_array();
        $res_2 = $res_2[0]["count(time)"];
    }
}
```

```

        $query1 = "delete from titik1 order by time asc limit
        ".(intval($res_1)-1000);
        $query2 = "delete from titik2 order by time asc limit
        ".(intval($res_2)-1000);

        $res1 = $this->db->query($query1);
        $res2 = $this->db->query($query2);
    }

    public function ambil_data($titik)
    {
        if($titik=="Tegangan"){
            $query1 = "SELECT * from titik1 ORDER
            BY time ASC";
            $res1 = $this->db->query($query1);
            $res1 = $res1->result_array();
            return $res1;
        }
        else if($titik=="Arus"){
            $query2 = "SELECT * from titik2 ORDER
            BY time ASC";
            $res2 = $this->db->query($query2);
            $res2 = $res2->result_array();
            return $res2;
        }
    }
}

```

Program sub.php

```
<html>
<head>
  <title>Line Charts and Options</title>
  <script language="javascript" type="text/javascript"
src="/dist/jquery.min.js"></script>
  <script language="javascript" type="text/javascript"
src="/dist/jquery.jqplot.min.js"></script>
  <link rel="stylesheet" type="text/css"
href="/dist/jquery.jqplot.css" />

  <script type="text/javascript"
src="/dist/plugins/jqplot.canvasTextRenderer.min.js"></script>
  <script type="text/javascript"
src="/dist/plugins/jqplot.canvasAxisLabelRenderer.min.js"></script>

</head>
<body>
<h1>
  <?php
echo $judul;

  ?>
</h1>
<div style="float: left; ">
<table border="1">
<tr><td>No</td><td align="center">Time</td><td
align="center">Nilai</td></tr>
<?php
  $nomer=1;
  $dibalik=1000;
  $grafik="[";
  for ($i=1; $i<=sizeof($data); $i++){
    $grafik=$grafik."[".$nomer."].".$data[$i-
1][["Nilai"]."].";
    echo "<tr>";
    echo "<td>".$dibalik."</td>";
```

```

        echo "<td style='padding-left:20px;padding-
right:20px;' align='center'>".$.data[typeof($data)-$i][ "Time" ]."</td>";
        echo "<td style='padding-left:20px;padding-
right:20px;' align='center'>".$.data[typeof($data)-$i][ "Nilai" ]."</td>";
        echo"</tr>";
        $nomer++;
        $dibalik=$dibalik-1;
    }
    $grafik=$grafik."]]";
?>
</table>
</div>
<div id="chartdiv" style="margin-left:330px; right; width: 1000px; "
></div>
<script type="text/javascript">
    $('#chartdiv').css("height", "600px");
    $('#chartdiv').css("width", "1000px");
    $('#chartdiv').css({position:"fixed", top:"50"});

    function plotData(_data,a){
        console.log(_data);
        if (a==1){
            $.jqplot('chartdiv', _data,
            { title:'Grafik Tegangan',
            axes: {xaxis: {label:'No',min:1,
max:1000},yaxis: {label:'Nilai',min:0, max:263 }},
            series:[ {color:'#5FAB78',lineWidth:1,markerOptions: {size:1}} ]
            });
            console.log(1);
        }
        if (a==2){
            $.jqplot('chartdiv', _data,
            { title:'Grafik Arus',
            axes: {xaxis: {label:'No',min:1,
max:1000},yaxis: {label:'Nilai',min:-15, max:15}},
            series:[ {color:'#5FAB78',lineWidth:1,markerOptions: {size:1}} ]
            });

```

```
console.log(1);
}
}
</script>
<?php
    if ($judul=="Tegangan"){
        $a=1;
        echo "<script
type='text/javascript'>plotData(\".$grafik.\",\".$a.\")</script>";
    }
    if ($judul=="Arus"){
        $a=2;
        echo "<script
type='text/javascript'>plotData(\".$grafik.\",\".$a.\")</script>";
    }
?>
</body>
</html>
```



Program tampil.php

```
<html>
<head>
<link rel="stylesheet" href="style.css">
<meta http-equiv="refresh" content="5" >
</head>
<body>
<div id="satu"><a href="index.php/welcome/ambil/Tegangan"><?php
echo $nilai1?></a></div>
<div id="dua"><a href="index.php/welcome/ambil/Arus"><?php echo
$nilai2?></a></div>
</body>
</html>
```

BIODATA PENULIS



Penulis bernama lengkap Gilang Dwi Pamungkas lahir pada tanggal 14 Februari 1993, sebagai anak kedua dari dua bersaudara. Penulis mengawali kegiatan pendidikan formal di SDS Kemala Bhayangkari 1 Surabaya, yang kemudian dilanjutkan di SMP Negeri 1 Surabaya, kemudian SMA Negeri 5 Surabaya dan pada tahun 2011 penulis diterima sebagai mahasiswa di jurusan Teknik Elektro ITS. Saat kuliah, penulis turut berpartisipasi sebagai asisten praktikum di bidang studi elektronika serta aktif pula dalam unit kegiatan mahasiswa workshop Elektro ITS dan Electrical Engineering Event 2014.

Email : gildw1@gmail.com