

BAB III

METODE PENELITIAN

3.1 Lokasi Penelitian

Objek yang akan diteliti pada penelitian ini adalah rangkaian regulator keluaran sistem pembangkit listrik tenaga surya yang bertempat di Laboratorium tegangan tinggi Universitas Pendidikan Indonesia. Laboratorium tersebut berlokasi di Jl. Setiabudi No. 229 Gd. 2 FPTK , UPI, Sukasari, Bandung, Jawa Barat 40154, Indonesia.

3.2 Perangkat Penunjang Penelitian

Penelitian yang dilakukan menggunakan beberapa perangkat yang menunjang agar penelitian ini tercipta secara utuh dan bekerja dengan sesuai harapan. Pada penyusunan laporan penelitian yang terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan adalah seperangkat *MacBook Air* dengan spesifikasi prosesor *Intel Core i5* 1.4 GHz, memori 4 GB 1600 MHz DDR3, kartu grafis *Intel HD Graphics* 5000 1536 MB, dan ruang penyimpanan data 128GB. Inti perangkat lunak yang digunakan adalah sistem operasi *OS X El Capitan Version 10.11.1*, *Microsoft Word for Mac*, dan *Mendeley Desktop Version 1.15.3*.

Perangkat yang digunakan dalam proses konfigurasi dan perancangan sistem yaitu satu set perangkat Raspberry pi 2 Model B, satu buah *Wireless Nano USB Adaptor* TL-WN725N, satu buah papan mikrokontroler Arduino Nano, dan dua buah *transceiver* NRF24L01.

Perangkat yang digunakan dalam proses kalibrasi di antaranya:

1. Multimeter digital, sebanyak tujuh multimeter yang digunakan memiliki merek dan tipe yang sama yaitu SANWA CD800a. Sedangkan pengukuran arus menggunakan tang ampere bermerek KYORITSU, SANFIX, dan menggunakan multimeter merek HELES UX 838-TR.



Gambar 1.1 Multimeter untuk kalibrasi

1. DC Power supply, bermerek shinyoku tipe ADC1212T dengan spesifikasi Input: 220V~ 50Hz 19VA Output: 1.5, 3, 4.5, 6, 7, 8, 9 12 1200mA 14.5VA(max)



Gambar 1.2 DC Power supply

3.3 Spesifikasi PLTS dan Alat Pengukur

PLTS yang ada pada Lab Tegangan Tinggi FPTK UPI memiliki tegangan keluaran sebesar 100 watt. Berhubung *solar cell* ini sudah sangat lama digunakan, sehingga efisiensi yang diperoleh dari PLTS adalah sebesar 30%. Sehingga nilai keluarannya sebesar ± 30 watt. Nilai ini merupakan nilai daya maksimum yang dapat dihasilkan oleh PLTS. Sebelum memilih spesifikasi yang cocok untuk

Regy Roy Hidayat Putra, 2018

RANCANG BANGUN SISTEM PENGAWASAN DAYA DAN ENERGI PADA PEMBANGKIT LISTRIK TENAGA SURYA MENGGUNAKAN RASPBERRY PI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

pengukuran PLTS, dilakukan terlebih dahulu pengukuran menggunakan alat ukur agar dapat dipastikan arus yang mengalir pada kondisi puncaknya (kondisi keadaan terang).



Gambar 1.3 Pelat nama solar panel

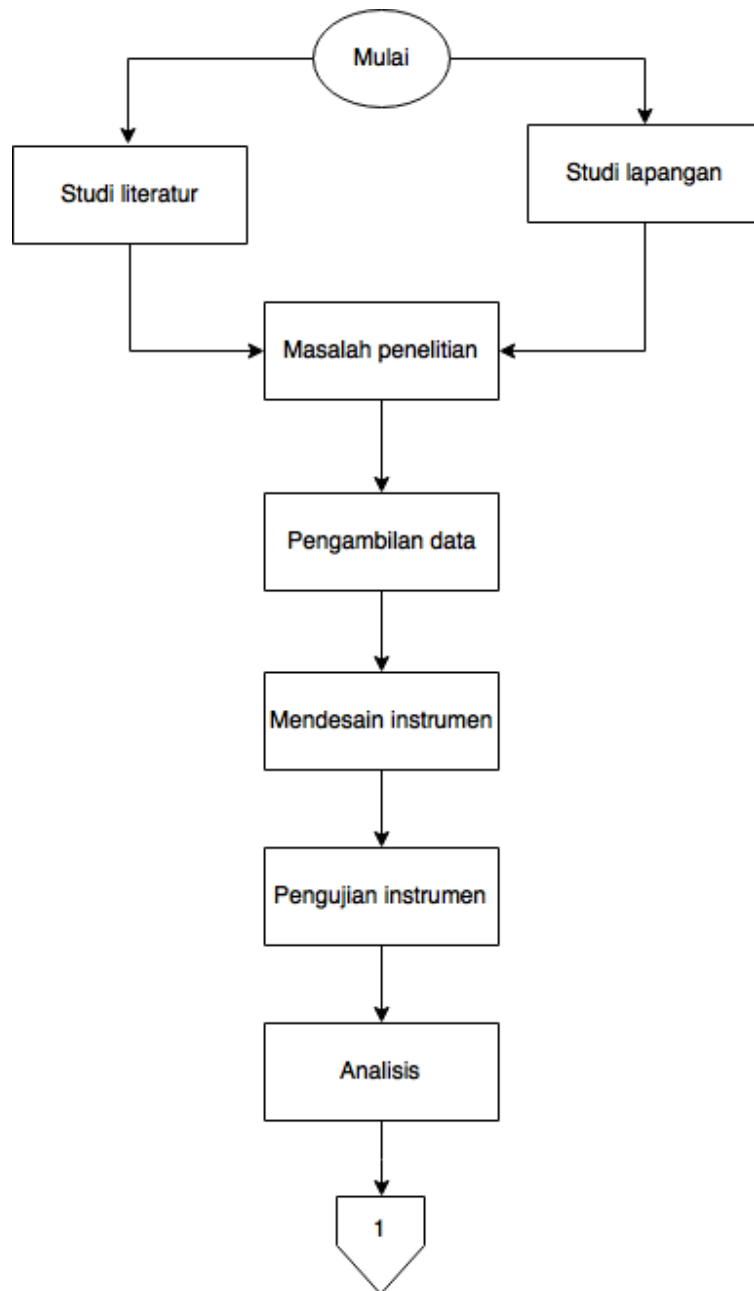
Selain itu mempelajari juga dari spesifikasi yang tersedia mengenai keluaran PLTS dari pelat nama yang tersedia pada solar panel seperti pada gambar 3.3. Setelah dipastikan nilai arus yang mengalir pada kondisi puncak, selanjutnya memilih perangkat yang sesuai yaitu dapat menggunakan sensor arus berukuran paling kecil 5A. Pemilihan sensor tegangan dengan memilih sensor tegangan yang dapat membaca <25 volt. Karena pada kondisi lapangan dan juga pelat nama yang tersedia tegangan yang terbaca adalah <20 Vdc.

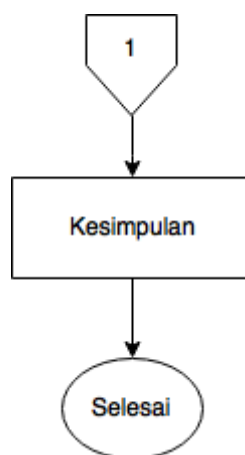
Server dan *node* yang bekerja membutuhkan daya untuk dapat beroperasi. Berdasar informasi dari web resmi Raspberry, bahwa Raspberry Pi 2 model B direkomendasikan menggunakan arus sebesar 1.8A, pendayaannya menggunakan 2A. Sedangkan Arduino Nano mengonsumsi daya untuk menghidupkan *main boardnya* diperlukan 19mA, sehingga menggunakan arus 1A pada pendayaannya agar modul lain yang terpasang pada Arduino dapat berfungsi. Karena Raspberry Pi dan Arduino Nano menggunakan tegangan 5V sebagai tegangan masukannya,

sehingga daya yang dibutuhkan adalah sebesar 9 watt untuk Raspberry Pi dan 5 watt untuk Arduino Nano. Pendayaan tersebut merupakan daya maksimal.

3.4 Alur Penelitian

Penelitian yang sistematis dan terarah terbentuk dengan bantuan diagram alir penelitian. Diagram alir penelitian mencakup langkah-langkah penelitian yang diawali dengan langkah pertama yaitu dengan melakukan studi literatur, menentukan masalah penelitian, pengambilan data, kemudian melakukan eksperimen, melakukan analisis dari hasil eksperimen, hingga menentukan kesimpulan.

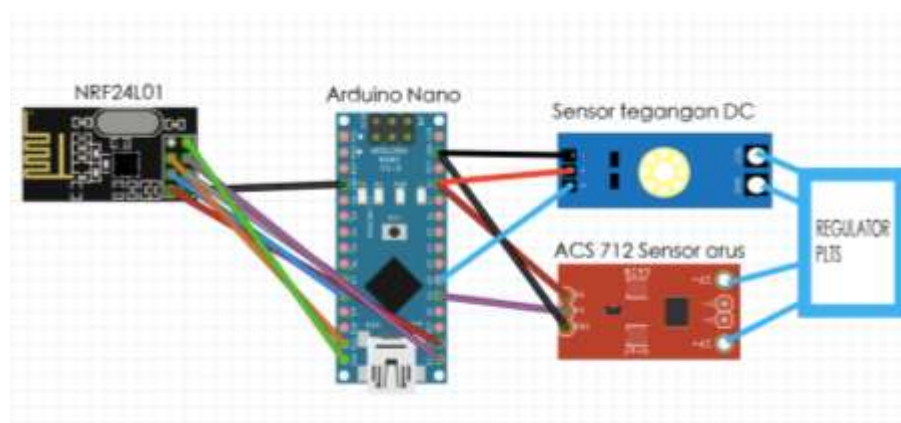




Gambar 1.4 Diagram Alir Penelitian

3.5 Prosedur Penelitian

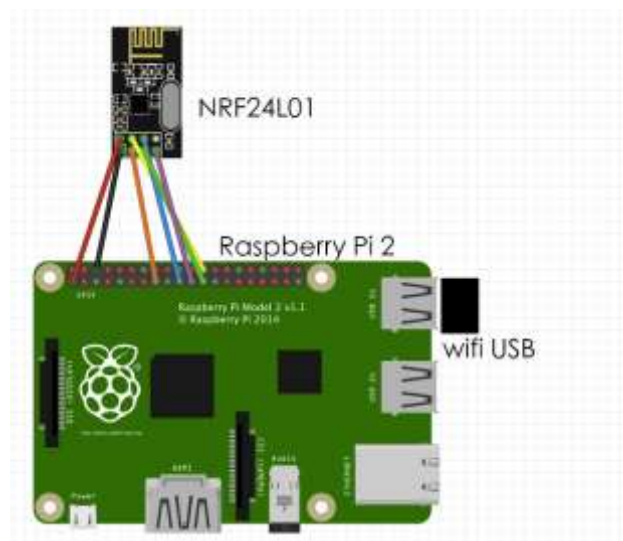
Sistem yang dirancang ini didesain dengan bertujuan agar alat dapat memonitor daya yang disuplai oleh sistem panel surya secara waktu nyata dan disimpan dengan teknik *logging system* pada basis data. Hasil keluaran data yang bersifat akuisisi data dapat diakses oleh pengguna melalui perangkat penjelajah halaman web. Pengguna yang akan mengakses sistem ini memerlukan koneksi lokal yang tersedia secara nirkabel. Sistem pengukuran ini terbagi menjadi dua bagian, yang pertama yaitu pada bagian sisi pengukuran dan yang kedua yaitu pada bagian sisi server.



Gambar 1.5 Koneksi *jumper* pada *node*

Sisi pengukuran dilakukan oleh perangkat mikrokontroler Arduino Nano yang diberikan daya masukan melalui hubungan mini-B USB. Sensor yang terdiri yaitu sensor arus ACS712 dan sensor tegangan. Kedua sensor dapat bekerja sekaligus. Gambar 3.5 merupakan rangkaian dari sensor-sensor, Arduino Nano,

NRF24L01 dan regulator PLTS yang saling terhubung. Rangkaian pengukuran beban, rangkaian untuk Arduino Nano, sensor-sensor, dan NRF24L01 sama halnya dengan rangkaian pada gambar 3.5. Pada bagian sisi sensor, pusatnya yaitu Arduino Nano menjalankan operasi perhitungan sederhana untuk menghitung daya. Operasi perhitungan ini terdapat pada program yang telah di-*upload* melalui Arduino IDE. Selain operasi perhitungan, terdapat pula program yang berfungsi untuk mengirim data hasil dari operasi perhitungan melalui perangkat *transceiver* NRF24L01.



Gambar 1.6 Koneksi *jumper* pada server

Raspberry Pi 2 model B diletakan pada sisi server. Perangkat ini didesain untuk memenuhi tujuan dari sistem yang diciptakan. Peran server yang dilakukannya di antaranya adalah sebagai sisi penerima data dari sistem atas data sensor yang dikirim. Peran server kedua adalah sebagai web server, data yang diterima akan ditampilkan pada halaman web secara *real-time* dengan menggunakan *software apache2*. Peran server ketiga sebagai *database server*, berfungsi sebagai log dari data yang ditampilkan. Data yang telah ditampilkan secara *real-time* kemudian disimpan pada *software database* yaitu *MySQL*. Untuk memudahkan konfigurasi dari basis data seperti membuat tabel dapat menggunakan bantuan manajemen *database PHPMyAdmin*. Peran server keempat yaitu sebagai DHCP server yang bekerja untuk memberikan alokasi alamat IP yang terhubung dengan jaringan nirkabel yang telah disediakan oleh sistem.

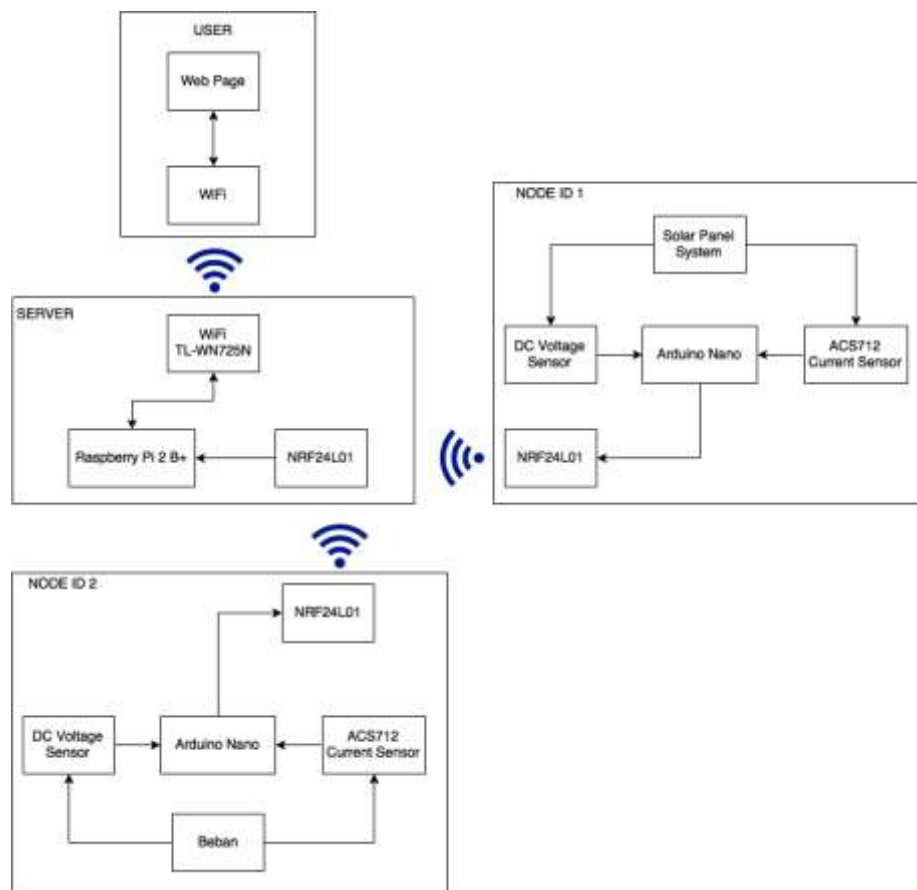
Peran kelima yaitu sebagai DNS server yang memudahkan pengguna untuk mengakses server dengan nama domain.

3.6 Diagram Blok Sistem

Perangkat Raspberry Pi 2 model B yang menjadi pusat dari sistem ini bekerja untuk menerima nilai dari bacaan sensor yang dikirim dari Arduino Nano. Data tersebut berupa nilai satuan watt yang terbaca oleh sensor tegangan DC dan sensor arus AC712. Sensor AC712 yang digunakan adalah dengan spesifikasi 20A (kapasitas 20 ampere) dan 5A (kapasitas 5 ampere); dan modul sensor tegangan yang dipilih memiliki batasan kerja pembacaan tegangan kurang dari 25 volt DC. ACS712 20A digunakan pada bagian *node* pembacaan daya PLTS. ACS712 5A digunakan pada bagian *node* pembacaan daya beban.

Tampilan data berupa grafik dari bacaan di bagian sensor, ditampilkan oleh perangkat Raspberry Pi 2 model B dengan bantuan program Highchart. Program ini merupakan skrip Javascript yang berguna untuk menampilkan nilai menjadi tampilan grafik. Skrip Javascript yang digunakan disisipkan pada berkas web utama (index.html) karena grafik akan ditampilkan pada web halaman utama. Kemampuan dari Highchart ini yang dapat bekerja menampilkan bacaan nilai secara *real-time* menjadikannya menjadi pilihan sebagai tampilan antarmuka di halaman web. Selain itu, data yang sebelumnya telah disimpan di dalam basis data dapat dilihat kembali per tanggal yang diinginkan.

Pada gambar 3.7 menampilkan blok diagram yang merupakan kesatuan perangkat sistem yang terdiri dari empat elemen yang terlibat, yaitu dua buah Arduino Nano yang masing-masing memiliki sensor arus dan sensor tegangan untuk mendapatkan nilai daya yang akan diawasi, sebuah Raspberry Pi 2 B yang menjadi perangkat pusat dari sistem ini, dan perangkat pengguna yang ingin mengakses Raspberry Pi 2 B, dan perangkat pengguna yang dapat menggunakan perangkat baik sebuah *Personal Computer* atau Ponsel. Salah satu atau kedua perangkat pengguna tersebut dapat mengakses Raspberry Pi 2 B dalam satu waktu atau dalam waktu yang bersamaan.



Gambar 1.7 Diagram Blok Sistem

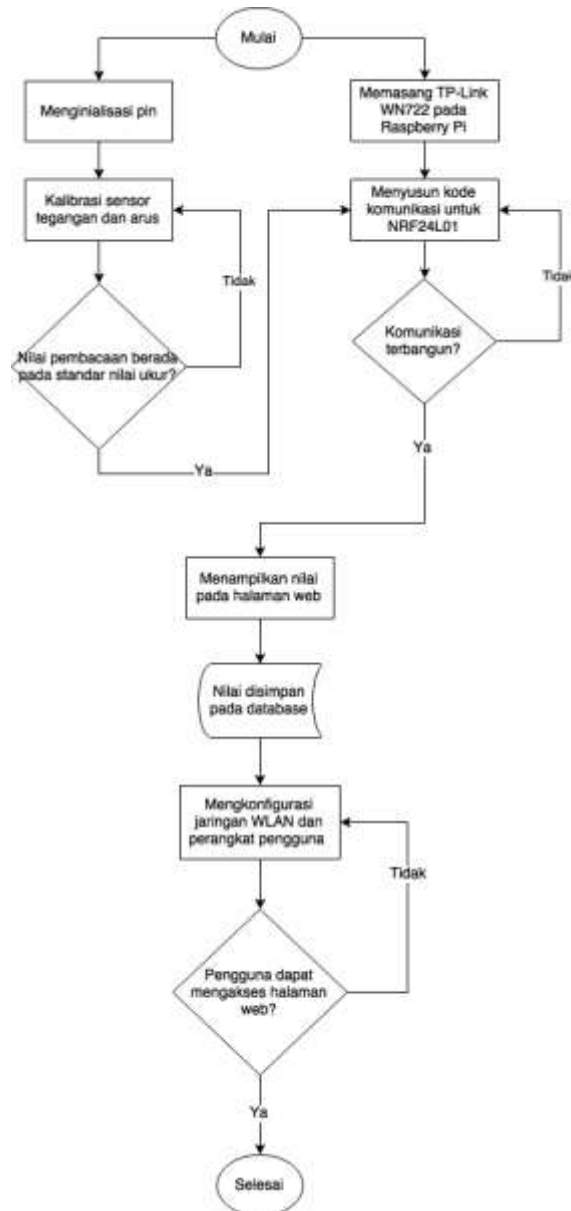
Bermula dari solar panel yang terdapat aliran daya ke baterai, nilai daya tersebut dibaca oleh sensor tegangan dan sensor arus. Nilai dari kedua bacaan sensor tersebut dioperasikan perkalian sehingga dihasilkan nilai daya pada Arduino Nano. *Node* yang tersedia sebanyak dua *node*, sehingga *node* lainnya digunakan untuk simulasi mengukur daya beban. Beban yang disimulasikan berupa sumber tegangan DC dan tahanan. Karena setiap *node* memiliki sensor arus dan tegangan DC.

Selanjutnya data yang didapatkan secara periodik selama satu detik ditransmisikan melalui *transceiver* NRF24L01 menuju bagian server yang memiliki NRF24L01. Perangkat Raspberry Pi 2 Model B menerima data daya tersebut dengan program skrip yang berbahasa pemrograman *python* maka dijalankan skrip tersebut sehingga data daya diterima dan langsung disimpan di program basis data. Selanjutnya perangkat Raspberry Pi 2 Model B ini menampilkan data tersebut ke dalam bentuk grafik dengan bantuan perangkat

lunak web server. Hingga akhirnya tampilan pada web server dapat diakses pada jaringan nirkabel *WiFi* dengan menggunakan penjelajah halaman web.

3.7 Perancangan Sistem

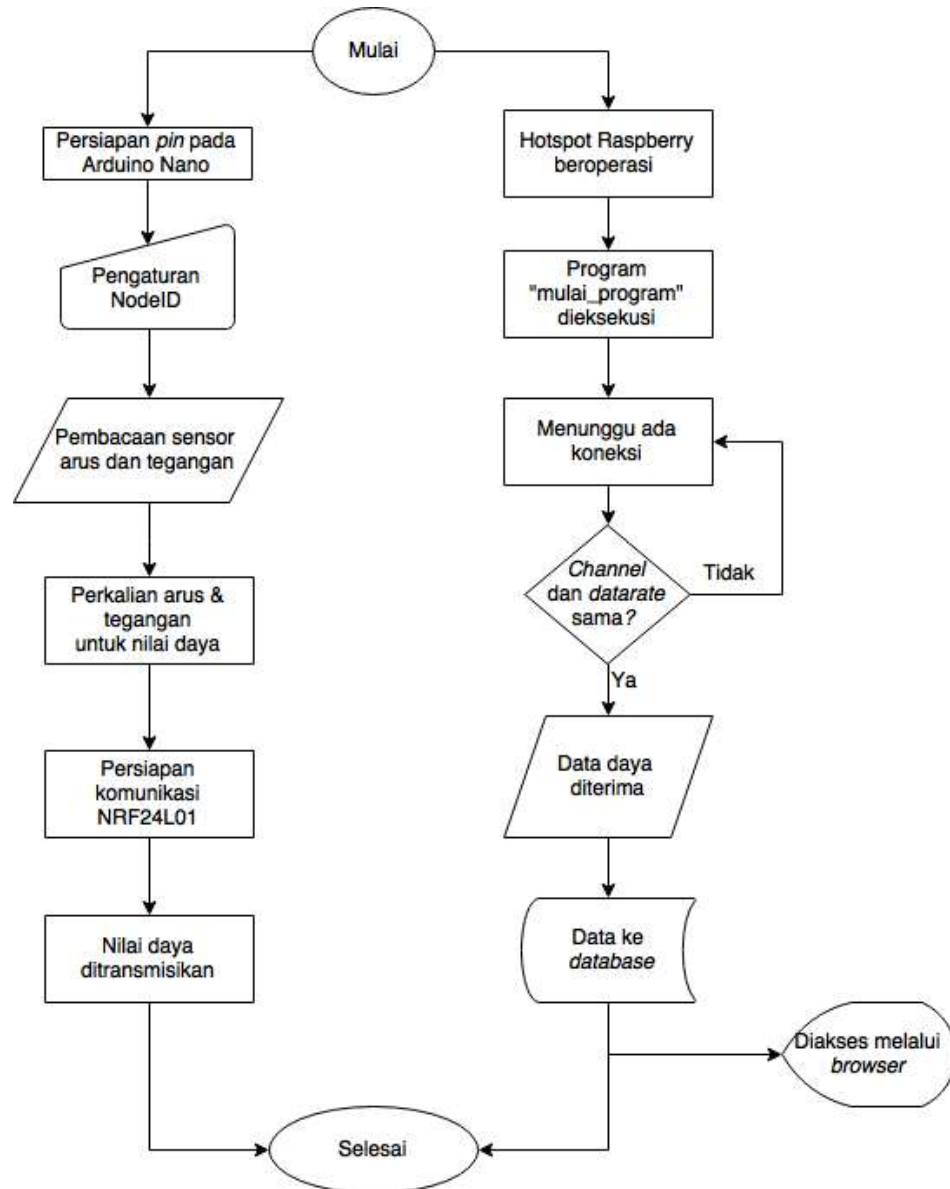
Proses perancangan atau dalam proses pemecahan masalah akan lebih mudah ketika terjadi galat maka diperlukan dibuat diagram alir desain sistem.



Gambar 1.8 Diagram alir sistem kerja sistem

Diawali dengan pemilihan perangkat yang terlebih dahulu untuk diprogram, perangkat Raspberry Pi 2 model B atau perangkat Arduino Nano terlebih dahulu. Pada program Arduino, menggunakan bahasa C sebagai bahasa

pemrogramannya. Terdapat modul yang tersedia di internet yang dapat digunakan untuk sensor dan metode kalibrasinya. Secara keseluruhan runtutan operasi yang didesain sesuai pada gambar 3.9.



Gambar 1.9 Diagram alir program keseluruhan sistem

Gambar 3.9 menunjukkan proses yang dilakukan pada saat sistem bekerja. *Node* dan server bekerja dalam satu waktu. Proses selesai ketika data telah masuk ke *database* setelah daya ditransmisikan dan dapat diakses oleh *browser*.

3.8 Desain Program Perangkat Lunak

Program yang akan diterapkan pada sistem terdiri dari beberapa bentuk program. Program-program tersebut berhubungan masing-masing. Program yang dibuat dibedakan berdasar titik peran pada jaringan sistem, yaitu server yang ada pada raspberry dan *node* yang ada pada masing-masing *node*.

1.8.1 Perangkat Lunak pada *Node*

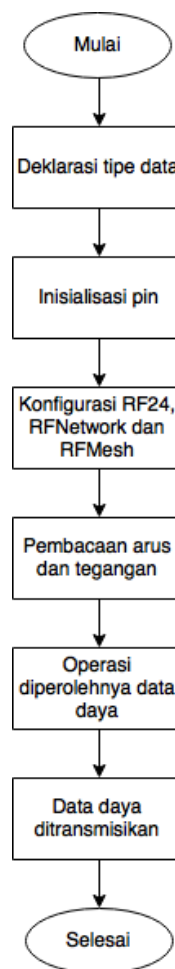
Node yang diprogram memiliki pembeda dari dengan yang lainnya. Hal ini bertujuan untuk membedakan antara satu *node* dengan *node* lainnya. Meskipun secara keseluruhan kode program pada masing-masing *node* identik, terdapat satu baris yang bekerja sebagai pembeda. Pada jaringan *mesh*, *node* diberi identitas dengan mendefinisikan kode NodeID.

```
#define nodeID 1
```

Gambar 1.10 NodeID

Pemrograman *node* dibuat untuk melaksanakan tugas mengukur tegangan dan arus pada satu waktu. Dimulai dari proses inialisasi *pin*, yaitu nomor *pin* berapa saja yang digunakan untuk pembacaan sensor arus dan tegangan pada Arduino Nano. Selain penentuan nomor *pin* untuk sensor, diperlukan juga penentuan nomor *pin* untuk modul komunikasi NRF24L01. Pin yang dapat diatur untuk modul komunikasi NRF24L01 hanya untuk konfigurasi *pin* CE (*Chip Enable*) dan CS (*Chip Select*). Selibhnya, menghubungkan *pin* sesuai dengan fungsi tetap dari masing-masing nomor *pin* di Arduino. Nomor *pin* yang dikonfigurasi sebelumnya harus dideklarasikan terlebih dahulu tipe datanya. NodeID ini mewakili alamat dari setiap *node*. NodeID yang dipilih adalah angka 1 dan 2. NodeID 0 dipakai untuk Raspberry Pi 2 B sebagai master. NodeID 1 digunakan untuk identitas *node* pengawas PLTS, sedangkan NodeID 2 digunakan untuk identitas *node* pengawas beban. NodeID 1 dan 2 diprogram untuk terintegrasi dengan Highchart sehingga jika NodeID selain angka 1 dan 2 maka akan mengakibatkan tidak munculnya grafik namun hanya nilainya saja yang tersimpan pada *database*.

Konfigurasi yang terdapat pada *node*, setelah berisi konfigurasi nomor *pin* yaitu konfigurasi mengenai bagaimana komunikasi dari modul NRF24L01 akan dibangun. Penggunaan modul program seperti RF24, RF24Network dan RF24Mesh digunakan. Model komunikasi antar perangkat menggunakan konsep *mesh* maka modul program *mesh* menjadi sangat penting karena RF24Mesh bergantung pada modul program RF24Network dan RF24. Pemilihan *channel mesh* diperlukan untuk mengenali titik yang merupakan bagian dari komunikasi *mesh*.



Gambar 1.11 Rancangan program sistem node

Nilai yang didapat dari bacaan sensor arus dan tegangan kemudian diolah dan dijadikan satu paket menjadi satuan daya. Sensor arus ACS712 bekerja dengan prinsip *effect hall* (Sulaiman, dkk. 2015) yaitu mengubah nilai analog menjadi nilai digital dengan memanfaatkan medan magnet sebagai media pengubah. Nilai analog yang diteruskan pada IC diubah menjadi tegangan DC

Regy Roy Hidayat Putra, 2018

RANCANG BANGUN SISTEM PENGAWASAN DAYA DAN ENERGI PADA PEMBANGKIT LISTRIK TENAGA SURYA MENGGUNAKAN RASPBERRY PI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

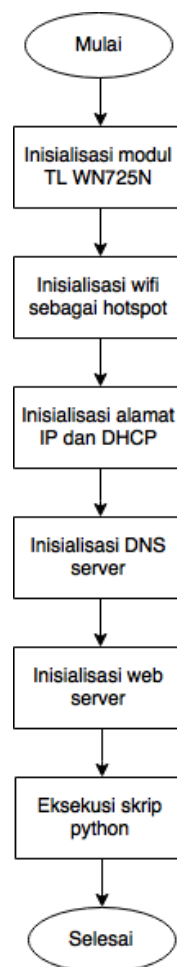
proporsional 0-5 volt. Berbeda dengan sensor tegangan yang bekerja berdasar prinsip pembagi tegangan.

Hasil inti keluaran dari program *node* selanjutnya siap dikirim melalui media NRF24L01 ke server. Secara runtut proses kerja dari program *node* terdapat pada gambar 3.11.

1.8.2 Perangkat Lunak pada Server

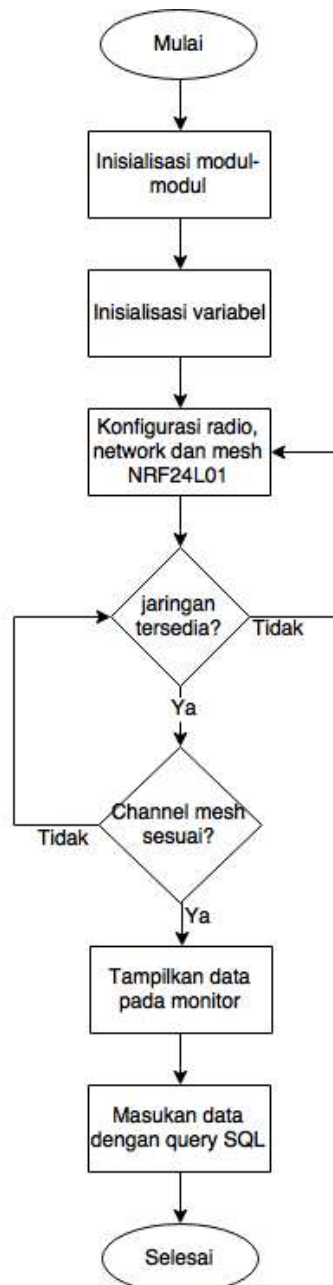
Raspberry yang digunakan akan diterapkan beberapa program untuk beroperasinya sistem pencatatan daya sehingga dapat diakses oleh pengguna menggunakan berbagai perangkat yang memiliki koneksi Wi-Fi dan *browser*.

Sistem perangkat lunak yang didesain terdiri dari tiga komponen, yaitu program komunikasi perangkat NRF24L01, program komunikasi untuk pengguna, dan program tampilan dari hasil pembacaan sensor. Secara keseluruhan sistem server yang mencakup ketiga program, ditunjukkan oleh gambar 3.11



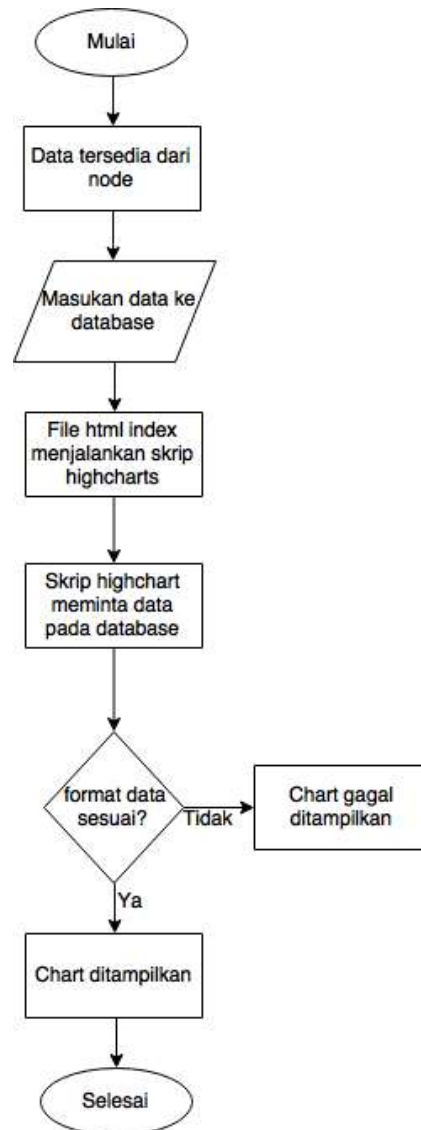
Gambar 1.12 Diagram alir sistem *server*

Program komunikasi perangkat NRF24L01 dibuat dengan bahasa pemrograman Python, dan dapat langsung dieksekusi melalui terminal di Raspbian. Gambar 3.13 mendeskripsikan alur kerja pada program komunikasi perangkat.



Gambar 1.13 Desain skrip program

Program untuk menampilkan hasil data yang dikirim oleh sensor yang berupa satuan daya, ditampilkan oleh Raspberry menjadi grafik dan hasil dari grafik tersebut dapat diakumulasikan per hari. Gambar 3.14 menunjukkan proses data diolah menjadi grafik.



Gambar 1.14 Diagram alir penampilan pada web

Penggunaan server web dilakukan ketika Raspberry pertama kali dinyalakan. Memodifikasi data pada *database* dapat langsung dilakukan tanpa mengganggu proses kerja server. Namun untuk melakukan perubahan setingan web server memerlukan proses memulai ulang program.

BAB IV

TEMUAN DAN PEMBAHASAN

3.1 Temuan Hasil Penelitian

Berdasarkan hasil penelitian dari eksperimen dalam mendesain sistem ini di Laboratorium tegangan tinggi FPTK UPI, diperoleh data sebagai berikut :

1. Hasil rancangan purwarupa sistem pencatatan daya listrik pada sel surya di Laboratorium tegangan tinggi FPTK UPI
2. Hasil rancangan dan pengolahan program-program pendukung sistem pencatatan daya listrik pada sel surya.
3. Hasil keluaran sistem daya listrik yang dapat diakses dalam jaringan lokal dengan menggunakan berbagai perangkat pengguna.

2.1.1 Hasil Perancangan Sistem Pencatatan Daya Listrik

Perangkat-perangkat yang saling terhubung antara server dan *node* dapat berkomunikasi dengan baik dan dapat mengirimkan data daya ke server.



Gambar 2.1 Raspberry Pi 2 B sebagai server

Server yang terdiri dari perangkat Raspberry Pi, modul NRF24L01, dan modul USB Wi-Fi. Gambar 4.1 menunjukkan gambar server yang terintegrasi antar ketiga perangkat tersebut.



Gambar 2.2 Arduino Nano sebagai *node*

Node yang merupakan pembaca nilai arus dan tegangan pada PLTS atau beban. Gambar 4.2 menunjukkan *node* yang terhubung dengan modul NRF24L01.



Gambar 2.3 *Node*

Node dan server yang telah dihubungkan dengan modul NRF24L01 telah siap dioperasikan. Rangka kotak penutup pada gambar 4.3 dan 4.4 dibuat sebagai purwarupa dari alat yang didesain. Terdapat kabel keluaran yang di dalamnya terhubung dengan sensor arus dan tegangan.



Gambar 2.4 Server

Pengujian sensor perlu dilakukan untuk menyesuaikan alat ukur yang akan diciptakan sesuai dengan standar. Sensor yang digunakan pada sistem ini terdapat dua buah yaitu sensor tegangan dan sensor arus DC berseri ACS712. Pemenuhan nilai standar pengukuran dilakukan dengan membandingkan sepuluh buah alat ukur (Multimeter).



Gambar 2.5 Pengujian nilai sensor pada Arduino IDE

Sepuluh alat ukur tersebut mengukur beban dan tegangan sama yang dijadikan sebagai acuan. Setelah didapatkan nilai standar dari multimeter, selanjutnya dibandingkan dengan sensor-sensor. Melalui program Arduino IDE, Arduino nano terhubung dengan komputer dilihat nilai pembacaannya melalui komunikasi serial seperti yang ditunjukkan pada gambar 4.5.



Gambar 2.6 Multimeter pengujian

Gambar 4.6 menunjukkan perangkat alat ukur yang dijadikan sebagai acuan untuk membandingkan nilai sensor bacaan tegangan. Alat ukur tersebut diatur untuk melakukan pembacaan tegangan saja.



Gambar 2.7 Rheostat

Rheostat digunakan sebagai beban. Beban yang dapat diatur nilainya dengan cara menggeser ke kiri dan ke kanan untuk mengubah nilai tahanannya.



Gambar 2.8 DC Power Supply

Tegangan dan arus yang diukur pada kalibrasi bersumber dari *power supply*.

Tabel 2.1 Hasil pengujian tegangan

| Alat Ukur Ke- | Nilai (V) |
|-------------------|-----------|
| 1 | 10.22 |
| 2 | 10.17 |
| 3 | 10.21 |
| 4 | 10.22 |
| 5 | 10.24 |
| 6 | 10.25 |
| 7 | 10.26 |
| Voltage DC Sensor | 10.14 |

Pada tabel 4.1 yaitu hasil pengukuran tegangan yang dilakukan dengan menggunakan tujuh buah multimeter. Nilai yang diperoleh dari masing-masing alat ukur berbeda, hal ini disebabkan dari alat ukurnya itu sendiri yang memiliki toleransi masing-masing walaupun ketujuh multimeter tersebut memiliki merek dan tipe yang sama.

Tabel 2.2 Hasil pengujian arus

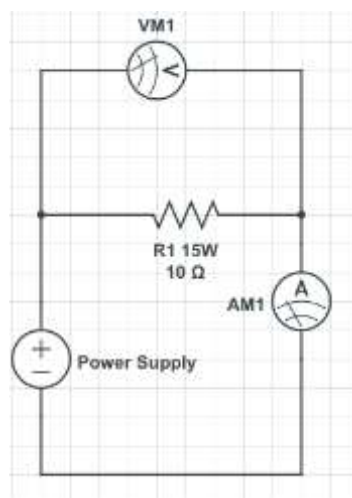
| Alat Ukur | Nilai (A) |
|---------------|-----------|
| 1 | 2.64 |
| 2 | 2.60 |
| 3 | 2.75 |
| 4 | 2.62 |
| 5 | 2.65 |
| Sensor ACS712 | 2.68 |

Pengujian yang dilakukan untuk mengkalibrasi sensor arus, menggunakan lima alat ukur. Tabel 4.2 menunjukkan nilai dari pengukuran alat ukur. Nilai berbeda tersebut bisa disebabkan oleh berbedanya alat ukur. Nilai arus yang didapatkan berasal dari beban yang diatur yaitu nilai rheostat yang mendekati 1Ω . Nilai yang terukur berada di kisaran 2.6 ampere.

Tabel 4.1 dan tabel 4.2 sensor arus dan sensor tegangan dibandingkan dengan sepuluh multimeter dengan objek beban dan tegangan yang sama.

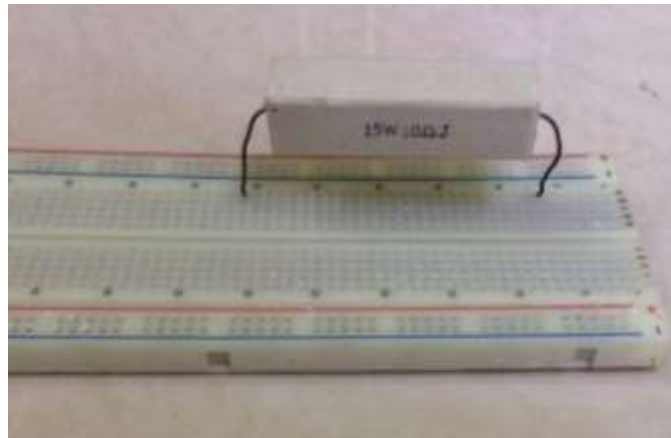
2.1.2 Pemasangan *Node* pada PLTS

Pemasangan sensor tegangan sama halnya dengan menggunakan multimeter, yaitu perlu memperhatikan *probe* positif dan negatif. Pemasangannya secara paralel terhadap rangkaian yang akan diukur tegangannya. Pemasangan sensor arus serupa dengan menggunakan multimeter untuk mengukur arus, dipasang secara seri. Hal ini juga berlaku pada pemasangan sensor di regulator PLTS.



Gambar 2.9 Rangkaian Simulasi

Pengujian sensor arus dibandingkan dua buah alat ukur. Objek yang diukur merupakan beban. Beban yang digunakan adalah dengan menggunakan *rheostat*. Nilai tahanan yang dipilih sebesar 5Ω . Tegangan yang digunakan pada *display* yang dipilih adalah sebesar 12V.



Gambar 2.10 Rangkaian Simulasi Beban

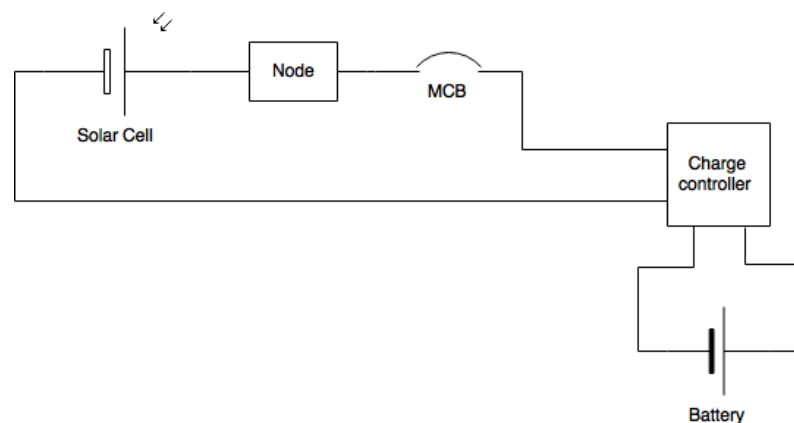
Gambar 4.10 merupakan beban simulasi. Beban yang dipilih untuk simulasi beban adalah sebuah resistor yang memiliki nilai 10Ω 15W.



Gambar 2.11 *Node* pada PLTS

Sama pada pengukuran arus dan tegangan pada simulasi beban, sensor tegangan dipasang secara paralel pada rangkaian regulator dan sensor arus dipasang secara seri di rangkaian regulator tepatnya pasang sebelum MCB. Regulator PLTS dicek terlebih dahulu keluaran nilai arus dan tegangannya untuk disesuaikan dengan kapasitas sensor yang akan digunakan. Gambar 4.11 merupakan *node* yang terintegrasi dengan PLTS.

Node yang dipasang pada PLTS diberikan identitas 1 pada kode pemrogramannya. *Node* yang dipasang beroperasi melalui daya eksternal, yaitu yang berasal dari *power bank* melalui kabel mini B USB.



Gambar 2.12 Rangkaian PLTS

Gambar 4.12 menunjukkan rangkaian dari sel surya menuju *charge controller* hingga baterai. Jarak *node* dengan server dapat berkomunikasi dengan baik pada jarak ± 10 meter.

2.1.3 Perancangan dan Pengolahan Program

Pemilihan perangkat lunak sistem perlu dilakukan karena setiap perangkat lunak memiliki fungsi kerja masing-masing. Perangkat lunak yang harus bekerja agar memberikan dukungan pada sistem bekerja di antaranya adalah *driver interface*, web server yang berfungsi sebagai penampil halaman web, *database server* untuk melakukan penyimpanan data, DHCP server untuk memberikan alamat IP kepada *user*, DNS server yang bekerja penerjemah alamat IP, dan *interface* laman web. Pastikan Raspberry mendapatkan koneksi internet untuk mempermudah pemasangan perangkat lunak.

A. Driver pada Raspberry

Agar perangkat tambahan dapat bekerja maka diperlukan memasang *drivernya* terlebih dahulu, dalam sistem ini perangkat tambahannya adalah TP-Link 725 N dan modul NRF24L01. Raspberry tidak bekerja secara *plug and play* untuk perangkat modul TP-Link ini, maka dengan manual dipasang *drivernya* dengan cara mengunduh berkasnya. Berkas *driver* tersebut disediakan pada forum Raspberrynya.

```
root@raspberrypi:/home/pi/driver# https://dl.dropboxusercontent.com/u/80256631/8188eu-20150130.tar.gz
```

Gambar 2.13 Berkas *driver*

Langkah selanjutnya berkas telah berhasil diunduh maka diperlukan mengekstrak berkas tersebut. Menggunakan perintah `tar xzf nama_file.tar.gz` kemudian memasukkan perintah `./install.sh` untuk memasang *driver* tersebut. Setelah selesai maka diperlukan untuk menjalankan kembali Raspberry. Gambar 4.13 merupakan alamat untuk berkas *driver* yang akan diunduh.

Sama halnya dengan perangkat NRF24L01 yang harus dipasang *driver*. Karena NRF24L01 akan dipadukan dengan pemrograman Python maka diperlukan library yang dapat bekerja dengan bahasa pemrograman tersebut. Melalui forum terbuka Github, dapat ditemukan *library* yang dapat bekerja antara NRF24L01.

B. Web Server

Laman web yang akan ditampilkan merupakan tugas dari web server, ketika *user* akan mengakses server melalui peramban web maka respons dari server adalah memberikan berkas html yang ditampilkan langsung pada peramban web yang melakukan permintaan. Web server yang dipilih untuk dipasang ke Web server adalah web server apache. Hal ini dipilih karena sesuai dengan kebutuhan pada sistem yang efektif dan efisien. Hanya menampilkan beberapa halaman saja dan kinerja dari web server ini efektif untuk menangani permintaan dari jumlah *user* yang sedikit.

Memasang web server pada Raspberry dengan menggunakan perintah `apt-get install apache2`.

Regy Roy Hidayat Putra, 2018

RANCANG BANGUN SISTEM PENGAWASAN DAYA DAN ENERGI PADA PEMBANGKIT LISTRIK TENAGA SURYA MENGGUNAKAN RASPBERRY PI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# apt-get install apache2
```

Gambar 2.14 Pemasangan web server

Konfigurasi yang telah diberikan pada Apache dapat segera digunakan. Hanya menyimpan berkas-berkas yang ingin ditampilkan pada direktori `/var/www/html` yang merupakan lokasi *root* dari sebuah laman web.

```
pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:/var/www/html# ls
bak      datax.php  index.html  json.php  test.php
data.php i.html    js          tanggal  Untitled.php
root@raspberrypi:/var/www/html#
```

Gambar 2.15 Lokasi *root* laman web

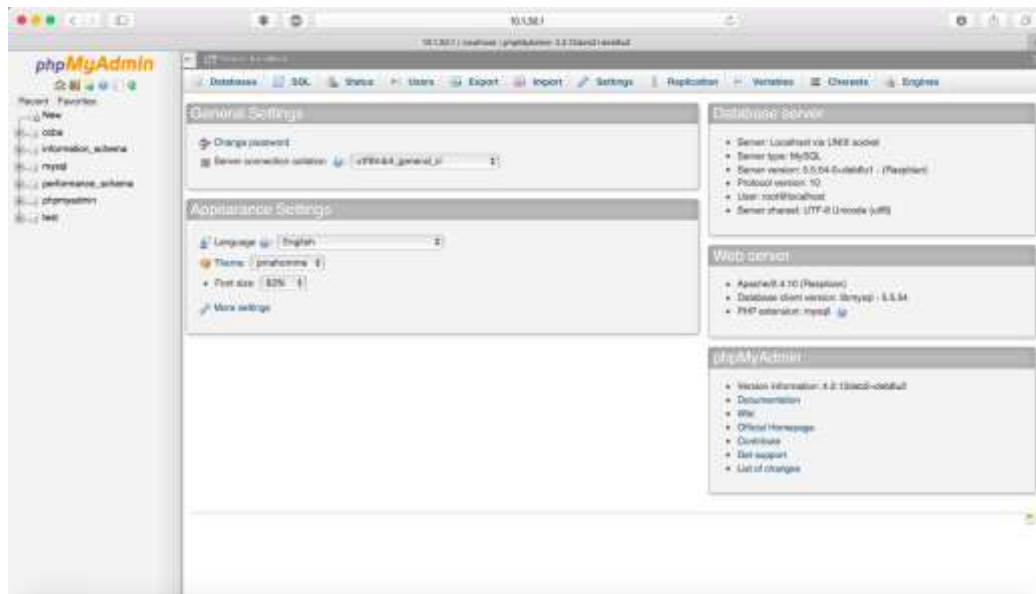
C. Database Server

Database server ini berfungsi untuk menyimpan catatan data yang disimpan dalam tabel. Sistem yang dirancang ini menggunakan *database* sebagai catatan daya dan waktu untuk ditampilkan. Pemasangan *database* server menggunakan metode yang sama seperti pemasangan web server karena *database* server yang digunakan, yaitu MySQL telah tersedia di *repository* Raspberry.

```
pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:~# apt-get install mysql-server phpmyadmin
```

Gambar 2.16 Pemasangan mysql dan phpmyadmin

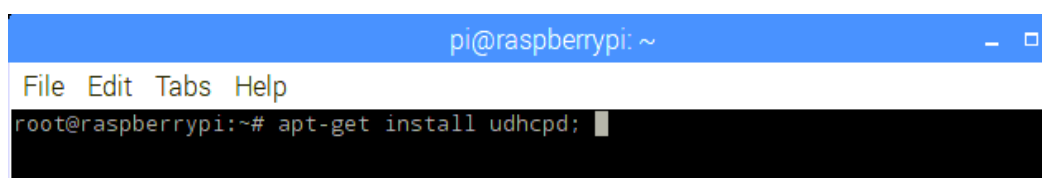
Agar lebih mudah dalam mengorganisir tabel dan isinya, dengan memasang phpmyadmin menjadi lebih mudah karena program tersebut akan dapat memodifikasi *database* melalui laman web. Ketika pemasangan berlangsung, akan meminta *password* baru sebagai *root* pada sistem *database*. Mengakses *database* melalui laman web dengan memasukkan alamat localhost.



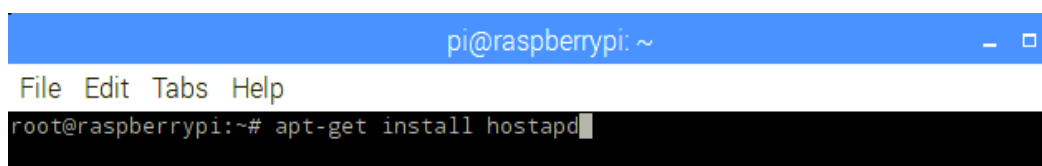
Gambar 2.17 Tampilan phpmyadmin pada laman web

D. Program *Hotspot*

Program agar menjadikan Raspberry *hotspot* adalah dengan cara memasang *hostapd* dan *udhcp*. *Hostapd* membuat modul wifi memancarkan sinyal wi-fi seakan-akan menjadi hotspot dan *udhcp* memberikan alamat IP untuk setiap perangkat yang terhubung dengan raspberry secara otomatis melalui jaringan wi-fi. Memasang *udhcp* dan *hostapd* hanya dengan memberikan perintah `apt-get install udhcpd` dan `hostapd`.



Gambar 2.18 Memasang udhcpd



Gambar 2.19 Memasang hostapd

Kedua program ini memiliki berkas konfigurasi masing-masing. *Udhcpd* memiliki konfigurasi pada lokasi `/etc/udhcpd.conf`.

E. DNS Server

Raspberry ketika akan diakses oleh user perlu mengarahkan browser ke alamat IP 10.1.30.1 namun akan lebih mudah jika hanya dengan mengetikkan nama domain saja. Alasannya karena akan lebih mudah diingat dengan nama dibandingkan dengan nomor alamat IP.

Pemanfaatan DNS server pada Raspberry dengan bantuan program DNS BIND9. Pemasangannya menggunakan *advance packaging tool* sama seperti pemasangan udhcpd, MySQL, dan program lainnya yaitu menggunakan perintah apt-get install.

F. Program “mulai_program”

Program ini berisi skrip yang menggunakan bahasa Python, bentuk dari program ini memiliki ekstensi .py sehingga untuk menjalankannya perlu memasang program Python. Python merupakan bahasa pemrograman yang sudah bawaan dari Raspberry, sehingga tidak perlu memasang program Python.

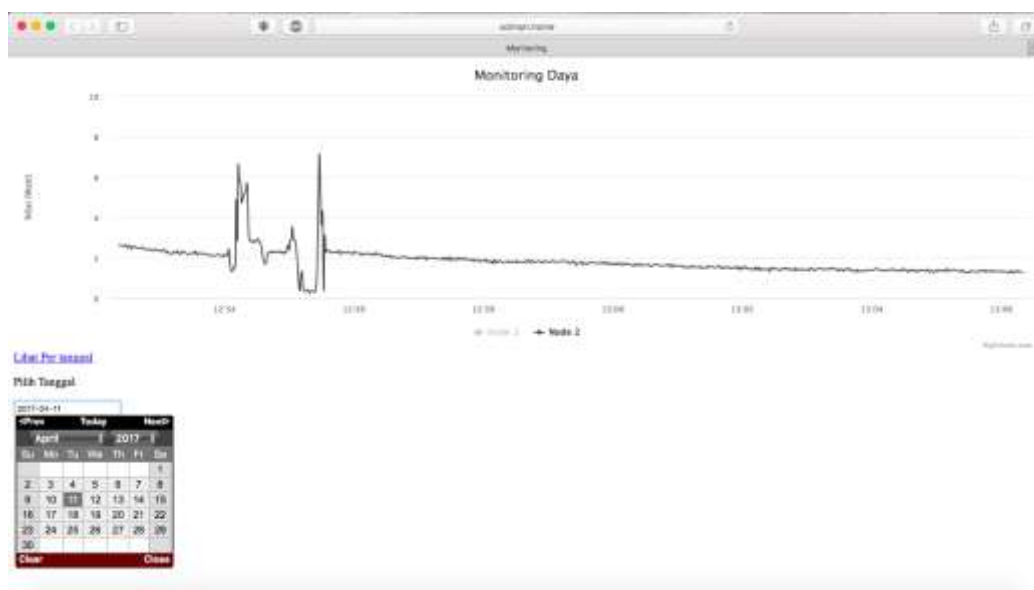
2.1.4 Hasil Sistem yang Dapat Diakses dalam Jaringan Lokal

Akses yang dilakukan oleh *user* kepada server setelah mendapatkan alamat IP maka selanjutnya *user* hanya memasukkan alamat web domain `http://solmon.home` di browser yang kemudian akan diarahkan ke alamat IP server yaitu 10.1.30.1. Ketika *node* yang aktif sedang mengukur, keluaran yang dapat diamati berupa grafik garis yang bergerak setiap detik.



Gambar 2.20 Tampilan halaman utama pada komputer

Kursor ketika diarahkan ke grafik akan menampilkan nilai yang terdiri dari nilai daya dan waktu. Kemudahan dalam menganalisis grafik dapat dipilih dari salah satu *node* saja yang ditampilkan. Tampilan web ini terdapat masukan per tanggal untuk melihat berapa jumlah daya yang terukur per hari.



Gambar 2.21 Pemilihan tanggal

Hasil dari pencatatan daya dapat dilihat per hari. Akumulasi pada hari tersebut dapat dilihat dengan operasi aritmetika pada MySQL. Pada gambar 4.22 contoh yang diambil adalah pada tanggal 10 April 2017.

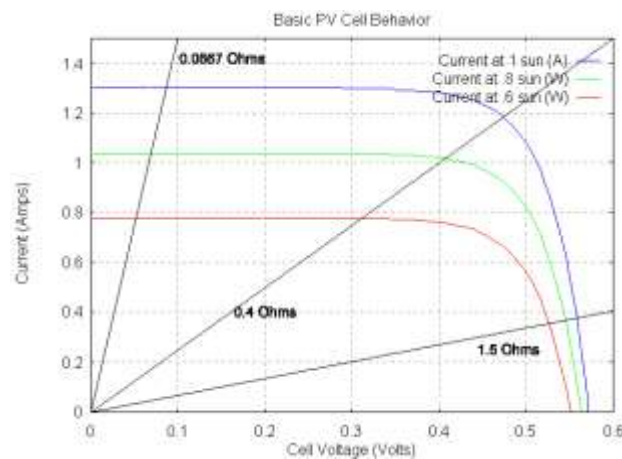
Data Energi Listrik

Pada tanggal : 2017-07-11
Total Energi : 13.3225 Wh

| Tanggal | Nilai (Watt) |
|---------------------|--------------|
| 2017-07-11 11:58:09 | 37.47 |
| 2017-07-11 11:58:10 | 37.28 |
| 2017-07-11 11:58:11 | 37.39 |
| 2017-07-11 11:58:12 | 37.69 |
| 2017-07-11 11:58:14 | 37.67 |
| 2017-07-11 11:58:15 | 37.41 |
| 2017-07-11 11:58:16 | 37.59 |
| 2017-07-11 11:58:17 | 37.32 |
| 2017-07-11 11:58:18 | 37.60 |
| 2017-07-11 11:58:19 | 37.48 |
| 2017-07-11 11:58:20 | 37.63 |
| 2017-07-11 11:58:22 | 37.69 |
| 2017-07-11 11:58:24 | 37.69 |
| 2017-07-11 11:58:26 | 37.06 |
| 2017-07-11 11:58:27 | 37.8 |
| 2017-07-11 11:58:31 | 37.69 |
| 2017-07-11 11:58:32 | 37.68 |
| 2017-07-11 11:58:33 | 37.89 |
| 2017-07-11 11:58:34 | 37.92 |
| 2017-07-11 11:58:35 | 38.03 |
| 2017-07-11 11:58:36 | 37.77 |
| 2017-07-11 11:58:37 | 38.07 |
| 2017-07-11 11:58:38 | 38.1 |
| 2017-07-11 11:58:39 | 37.66 |

Gambar 2.22 Tampilan hasil dalam waktu setengah jam

Energi yang dihasilkan oleh PLTS dalam waktu setengah jam adalah sebesar 13.33Wh seperti pada gambar 4.22. Hal ini dipengaruhi intensitas cahaya yang tersedia dan nilai daya optimal pada tegangan 12V. Hasil tersebut diperoleh pada pengukuran pada pukul 12.00-12.30.



Gambar 2.23 Karakteristik panel surya

Berdasar gambar 4.23, karakteristik dari sebuah sel surya memiliki daya keluaran maksimal. Pemilihan beban perlu diperhatikan agar mendapatkan masukan daya yang besar. Beban yang dipasang secara langsung dengan keluaran panel surya perlu ditetapkan seideal mungkin untuk mendapatkan daya masukan yang optimal.

2.1.5 Hasil Perancangan Sistem Pencatatan Daya Listrik

Setelah melakukan pengujian, data hasil pengukuran oleh multimeter dibuat ke dalam tabel untuk dicari nilai rata-rata dan standar penyimpangan. Perhitungan dilakukan dengan menggunakan Microsoft Excel untuk mendapatkan statistik deskriptif yang mencakup kedua nilai yang dicari. Tabel 4.3 dan 4.5 menunjukkan hasil perhitungan oleh bantuan program Microsoft Excel yang diolah dari tabel pengujian tegangan.

Tabel 2.3 Statistik deskriptif pengujian tegangan

| | |
|---------------------------|------------|
| <i>Mean</i> | 10.22 |
| <i>Standard Deviation</i> | 0.02992053 |
| <i>Minimum</i> | 10.17 |
| <i>Maximum</i> | 10.26 |
| <i>Sum</i> | 71.57 |
| <i>Count</i> | 7 |
| <i>Error</i> | 0.005% |
| <i>Smallest(1)</i> | 10.17 |

Mencari nilai rata-rata secara manual dapat dilakukan dengan menggunakan persamaan 4.1

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{N} \dots\dots\dots (4.1)$$

Sehingga nilai rata-ratanya dari ke tujuh multimeter adalah :

$$\bar{x} = \frac{10.22 + 10.17 + 10.21 + 10.22 + 10.24 + 10.25 + 10.26}{7}$$

$$\bar{x} = \frac{71.57}{7} = 10.22$$

Selanjutnya untuk menghitung standar penyimpangan dapat dibuat tabel tambahan untuk mempermudah perhitungan.

Tabel 2.4 Tabel perhitungan statistik pengujian tegangan

| Alat Ukur | Nilai (V) = x | $x - \bar{x}$ | $(x - \bar{x})^2$ |
|--------------------|-----------------|---------------|-------------------|
| 1 | 10.22 | -0.004285714 | 1.83673E-05 |
| 2 | 10.17 | -0.054285714 | 0.002946939 |
| 3 | 10.21 | -0.014285714 | 0.000204082 |
| 4 | 10.22 | -0.004285714 | 1.83673E-05 |
| 5 | 10.24 | 0.015714286 | 0.000246939 |
| 6 | 10.25 | 0.025714286 | 0.000661224 |
| 7 | 10.26 | 0.035714286 | 0.00127551 |
| Mean (\bar{x}) | 10.22 | Sum | 0.0054 |

Mencari standar penyimpangan dapat menggunakan persamaan 4.2

$$s = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}} \dots\dots\dots (4.2)$$

Sehingga nilai standar penyimpangannya adalah :

$$s = \sqrt{\frac{0.0054}{6}} = 0.0299$$

Maka nilai standar pada hasil pengukuran adalah 10.22 ± 0.0299 . *Error* yang diperoleh antara nilai dari bacaan sensor dengan alat ukur adalah:

$$Error = \frac{|nilai\ bacaan\ sensor - nilai\ multimeter|}{nilai\ bacaan\ sensor}$$

$$Error = \frac{|10.14 - 10.19|}{10.14} \times 100\%$$

$$Error = 0.005\%$$

Menggunakan metode yang sama, nilai yang diperoleh pada hasil pengujian arus menggunakan persamaan 4.1 untuk mencari nilai rata-rata dan persamaan 4.2 untuk mencari nilai standar penyimpangan.

Tabel 2.5 Statistik deskriptif pengujian sensor arus

| | |
|---------------------------|-----------|
| <i>Mean</i> | 2.652 |
| <i>Standard Deviation</i> | 0.0580517 |
| <i>Minimum</i> | 2.6 |
| <i>Maximum</i> | 2.75 |
| <i>Sum</i> | 13.26 |
| <i>Count</i> | 5 |

Maka nilai rata-rata dari hasil pengukuran arus adalah :

$$\bar{x} = \frac{2.64 + 2.6 + 2.75 + 2.62 + 2.65}{5}$$

$$\bar{x} = \frac{13.26}{5} = 2.652$$

Pencarian nilai penyimpangan standar menggunakan tabel tambahan.

Tabel 2.6 Tabel perhitungan statistik pengujian sensor arus

| Alat Ukur | Nilai (l) = x | $x - \bar{x}$ | $(x - \bar{x})^2$ |
|--------------------|-----------------|---------------|-------------------|
| 1 | 2.64 | -0.01 | 0.0001 |
| 2 | 2.6 | -0.05 | 0.0025 |
| 3 | 2.75 | 0.1 | 0.01 |
| 4 | 2.62 | -0.03 | 0.0009 |
| 5 | 2.65 | 0 | 0 |
| Mean (\bar{x}) | 2.652 | Sum | 0.01348 |

Nilai penyimpangan standarnya menjadi :

$$s = \sqrt{\frac{0.01348}{4}} = 0.0580517$$

Maka nilai standar pada hasil pengukuran adalah 2.652 ± 0.00580517 .

Sehingga untuk *error* yang didapatkan dari alat ukur adalah:

$$Error = \frac{|nilai\ bacaan\ sensor - nilai\ multimeter|}{nilai\ multimeter}$$

$$Error = \frac{|2.68 - 2.6527|}{2.68} \times 100\%$$

$$Error = 0.010\%$$

Berdasarkan kedua tabel pengujian, sensor tegangan dan arus menghasilkan pengukuran, pembacaan dari nilai sensor memiliki *error* yang kecil. Sehingga dapat disimpulkan sensor arus dan sensor tegangan bekerja dengan normal.

2.1.6 Hasil Perancangan dan Pengolahan Program

Program pendukung dari sistem ini berada pada di sisi server. Dimulai dari pemasangan *driver* NRF24L01 dan TP-Link 725N. Karena menggunakan program Python dalam pengoperasian sistem di Raspberry Pi maka *driver* yang dipilih agar dapat berkomunikasi NRF24L01 harus berbasis program Python juga *driver*-nya. Menggunakan bantuan program Boost.Python. Program ini bekerja untuk mengaktifkan operasi antara bahasa pemrograman C++ dan bahasa pemrograman Python. Hal ini dilakukan karena kebanyakan *library* yang tersedia di internet menggunakan bahasa pemrograman C++.

Penggunaan bahasa pemrograman Python memiliki keunggulan yang paling memudahkan pemrogram dalam membuat program dengan menggunakan Python, setelah skrip selesai ditulis kemudian disimpan dengan ekstensi Python (misal pada sistem ini adalah programku.py) skrip tersebut dapat langsung dijalankan tanpa melakukan proses kompilasi seperti pada bahasa pemrograman C++.

```

root@raspberrypi:/home/pi/Desktop/RF24Mesh-master 3/pyRF24Mesh# mulai_program
Memulai aplikasi
===== SPI Configuration =====
CSN Pin      = CE0 (PI Hardware Driven)
CE Pin       = Custom GPIO22
Clock Speed  = 8 Mhz
===== NRF Configuration =====
STATUS       = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
RX_ADDR_P0-1 = 0xffffffff 0xffffffff3c
RX_ADDR_P2-5 = 0x33 0xce 0x3e 0xe3
TX_ADDR      = 0xffffffff3ecc
RX_PW_P0-6   = 0x20 0x20 0x20 0x20 0x20 0x20
EN_AA        = 0x3e
EN_RXADDR    = 0x3f
RF_CH         = 0x4c
RF_SETUP     = 0x27
CONFIG       = 0x0f
DYNPD/FEATURE = 0x3f 0x04
Data Rate    = 250KBPS
Model        = nRF24L01+
CRC Length   = 16 bits
PA Power     = PA_MAX
P= 0.0 Node : 1

P= 0.0 Node : 1

```

Gambar 2.24 Respons dari server

Regy Roy Hidayat Putra, 2018

RANCANG BANGUN SISTEM PENGAWASAN DAYA DAN ENERGI PADA PEMBANGKIT LISTRIK TENAGA SURYA MENGGUNAKAN RASPBERRY PI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Gambar 4.24 menunjukkan konfigurasi SPI dan NRF. Ketika ada *node* yang terhubung dengan server melalui NRF. *Node* yang terhubung pada server adalah memiliki identitas 1. Setiap detiknya terdapat pengulangan data yang dikirim dari *node*. NRF24L01 bekerja pada frekuensi 2.400 hingga 2.525GHz. Pengaturan frekuensi dapat diatur dengan cara mengubah nilai *channel* yang digunakan. Berdasar *datasheet* Nordic Semiconductor yang merupakan produsen dari NRF24L01, formula untuk mengatur frekuensi yang digunakan adalah dengan menggunakan persamaan 4.3.

$$F_0 = 2400 + RF_CH [MHz] \dots\dots\dots(4.3)$$

Keterangan :

F_0 = Frekuensi yang akan diatur

RF_CH = *Channel* frekuensi yang diatur pada NRF

Channel yang dipilih bernilai 76, sehingga Frekuensi yang digunakan pada NRF24L01 adalah 2.476 GHz.

```

root@raspberrypi:/home/pi/Desktop/RF24Mesh-master 3/pyRF24Mesh# mulai_program
Memulai aplikasi
===== SPI Configuration =====
CSN Pin      = CE0 (PI Hardware Driven)
CE Pin       = Custom GPIO22
Clock Speed  = 8 Mhz
===== NRF Configuration =====
STATUS       = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
RX_ADDR_P0-1 = 0xffffffff 0xffffffff3c
RX_ADDR_P2-5 = 0x33 0xce 0x3e 0xe3
TX_ADDR      = 0xe7e7e7e7e7
RX_PW_P0-6   = 0x20 0x20 0x20 0x20 0x20 0x20
EN_AA        = 0x3e
EN_RXADDR    = 0x3f
RF_CH         = 0x4c
RF_SETUP     = 0x27
CONFIG        = 0x0f
DYNPD/FEATURE = 0x3f 0x04
Data Rate    = 250KBPS
Model        = nRF24L01+
CRC Length   = 16 bits
PA Power     = PA_MAX

```

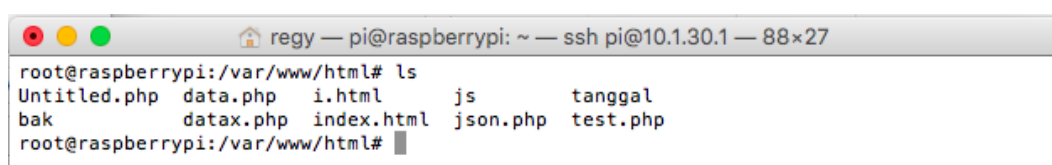
Gambar 2.25 Tidak ada respons dari *node*

Gambar 4.20 menunjukkan ketika program `mulai_program` dijalankan namun tidak ada satu pun *node* yang terhubung dengan server pada jaringan NRF. Tampilan tersebut menunggu hingga ada *node* yang terhubung. Hal ini dapat disebabkan karena tidak sesuainya konfigurasi antara *node* dengan server atau

memang tidak ada *node* yang tersedia. Konfigurasi yang harus sama antar kedua NRF ini adalah *channel* dan *Data Rate* yang digunakan.

A. Web Server

Apache merupakan program web server yang bersifat *open source* yang dapat digunakan pada sistem operasi Raspbian. Setelah melakukan pemasangan Apache, hanya dengan menyimpan berkas-berkas yang akan dimuat perlu disimpan di lokasi *directory* `/var/www/html`. Tidak memerlukan konfigurasi khusus untuk menampilkan berkas-berkas tersebut, cukup mengetikkan alamat *localhost* atau alamat IP dari web server itu sendiri.



```

root@raspberrypi:~$ ssh pi@10.1.30.1
root@raspberrypi:~$ cd /var/www/html
root@raspberrypi:/var/www/html# ls
Untitled.php  data.php  i.html    js         tanggal
bak          datax.php index.html json.php   test.php
root@raspberrypi:/var/www/html#

```

Gambar 2.26 Berkas yang akan ditampilkan

B. Database Server

Penggunaan MySQL pada sistem pencatatan daya ini dapat bekerja dengan baik pada server. MySQL dapat bekerja pada bahasa pemrograman Python sesuai dengan yang dikemukakan oleh Kofler (2005, hlm. 7) “*There are quite a number of APIs (application programming inter- faces) and libraries for the development of MySQL applications. For client programming you can use, among others, the languages C, C++, Java, Perl, PHP, Python, and Tcl.*”.

```

while network.available():
    header, N = network.read(9)
    ms, number = unpack('hh',bytes(N))
    print('P=', ms/100, 'Node :', abs(mesh.getNodeID(header.from_node)), oct(header.from_node), '\n')
    cursor.execute("INSERT into sensor (id,nilai) values (%s,%s)", [int(header.from_node),format(ms/100)])
    db.commit()

```

Gambar 2.27 Query pada program Python

Berpusat pada satu *database* yaitu *database* “coba” pada sistem pencatatan daya ini. *Database* coba berisi satu tabel dengan kolom waktu, nilai identitas node, dan nilai satuan daya.

C. Program *Hotspot*

Program *Hotspot* terdiri dari program `udhcpd` yang mengatur pengalamanan alamat IP kepada setiap perangkat yang terhubung pada jaringan server. Selain itu, program `hostapd` yang mengatur modul wi-fi pada server dapat memancarkan sinyal wi-fi yang dapat diakses.

```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/udhcpd.conf
# Sample udhcpd configuration file (/etc/udhcpd.conf)
# The start and end of the IP lease block
start 10.1.30.2 #default: 192.168.0.20
end 10.1.30.10 #default: 192.168.0.254
# The interface that udhcpd will use
interface wlan0 #default: eth0
# The maximim number of leases (includes addresssed reserved
# by OFFER's, DECLINE's, and ARP conficts
#max_leases 254 #default: 254

```

Gambar 2.28 Konfigurasi udhcp

Pada gambar 4.27 terdapat konfigurasi untuk memberikan alamat IP yang akan dialokasikan. Pada kasus ini diberikan alamat untuk *client* atau *user* dari 10.1.30.2 sampai 10.1.30.10. Sebanyak delapan perangkat saja yang diatur agar memberikan alamat IP secara otomatis ketika perangkat tersebut terhubung ke jaringan server.

Konfigurasi untuk `hostapd` berada pada berkas `/etc/hostapd/hostapd.conf`. Konfigurasi ini memuat *interface* mana yang akan digunakan dan *ssid* yaitu nama *hotspot* yang akan digunakan. Hanya kedua parameter itu saja yang dimodifikasi.

```

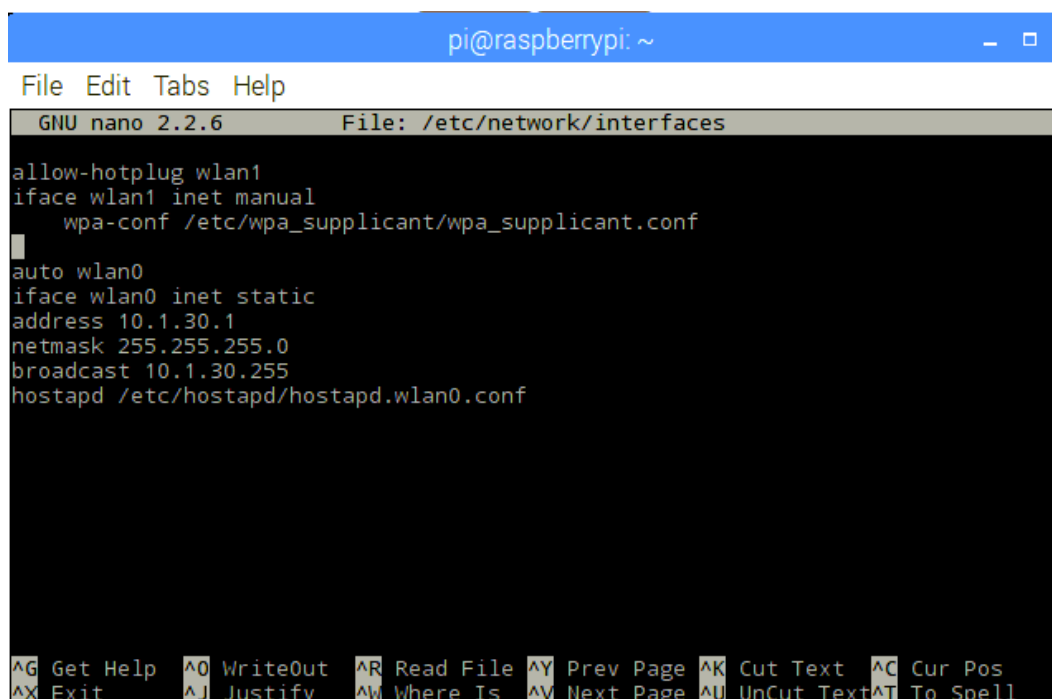
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/hostapd/hostapd.wlan0.conf
interface=wlan0
ssid=Monitoring
channel=1
ieee80211n=1

```

Gambar 2.29 Konfigurasi hostapd

Pelengkap dari konfigurasi `hostapd` berada pada `/etc/network/interfaces` yang merupakan konfigurasi jaringan pada sistem operasi UNIX. Tambahkan

aturan “auto wlan0 iface wlan0 inet static address 10.1.30.1 netmask 255.255.255.0 broadcast 10.1.30.255 hostapd /etc/hostapd/hostapd.wlan0.conf”



```

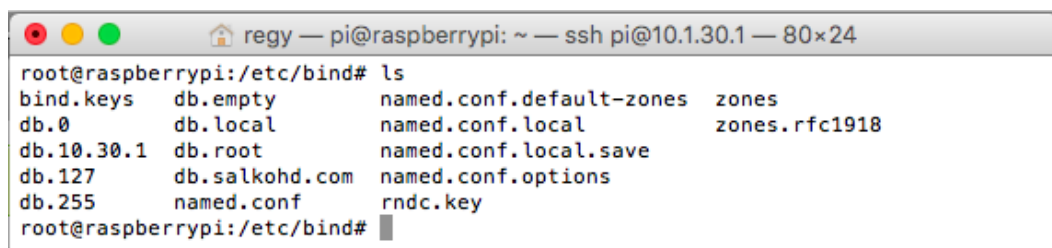
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/network/interfaces
allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
auto wlan0
iface wlan0 inet static
address 10.1.30.1
netmask 255.255.255.0
broadcast 10.1.30.255
hostapd /etc/hostapd/hostapd.wlan0.conf
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 2.30 Konfigurasi pada *network*

D. DNS Server

DNS Server pada Raspberry menggunakan program BIND, berkas konfigurasinya tersedia pada lokasi *directory* /etc/bind. BIND ini mengubah alamat IP server menjadi nama domain. Hal ini dilakukan untuk memudahkan pengguna untuk mengakses server yang hanya dengan memasukkan alamat domain `http://solmon.home` yang merupakan terjemahan alamat IP dari 10.1.30.1.



```

regy — pi@raspberrypi: ~ — ssh pi@10.1.30.1 — 80x24
root@raspberrypi:/etc/bind# ls
bind.keys  db.empty      named.conf.default-zones  zones
db.0       db.local      named.conf.local          zones.rfc1918
db.10.30.1 db.root       named.conf.local.save
db.127     db.salkohd.com named.conf.options
db.255     named.conf    rndc.key
root@raspberrypi:/etc/bind#

```

Berkas konfigurasi yang dibutuhkan adalah tiga berkas utama yaitu `db.10.30.1`, `db.salkohd.com` dan `named.conf.local`. Kedua file bernama `db` merupakan *database* dari alamat IP yang akan diterjemahkan ke dalam nama domain.

Regy Roy Hidayat Putra, 2018

RANCANG BANGUN SISTEM PENGAWASAN DAYA DAN ENERGI PADA PEMBANGKIT LISTRIK TENAGA SURYA MENGGUNAKAN RASPBERRY PI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

E. Program “mulai_program”

```

GNU nano 2.2.6 File: programku.py

#database stuffs
db = pymysql.connect("localhost","root","jkregy","coba")
cursor = db.cursor()
today = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
-

```

Gambar 2.31 Koneksi Python dengan MySQL

Skrip pada program yang dibuat berisi komunikasi dengan *node*, mendapatkan nilai dari *node*, hingga memasukkan nilai daya dari *node* ke *database*. Karena Python ini fleksibel memiliki keunggulan dapat memasang modul yang dibutuhkan. Python mendukung MySQL secara langsung, perintah *query* MySQL dapat ditulis pada skrip python. Supaya memudahkan dalam mengeksekusi programku.py maka dibuat skrip yang dijalankan pada terminal dengan mengetikkan “mulai_program”. Skrip yang dibuat merupakan perintah pada terminal untuk mengeksekusi programku.py. Dibandingkan memasukkan perintah “python3 programku.py” dengan lokasi *directory* aktif pada terminal maka pembuatan skrip “mulai_program” lebih efisien karena dapat dieksekusi pada lokasi *directory* mana pun.

2.1.7 Hasil Sistem yang Dapat Diakses dalam Jaringan Lokal

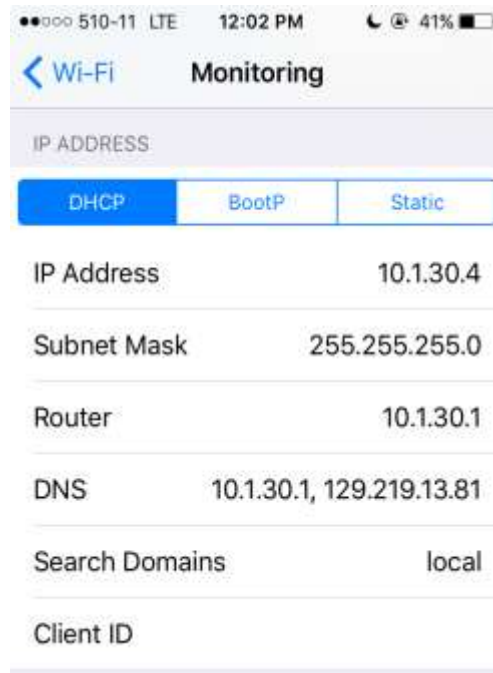
Perangkat yang akan mengakses sistem untuk mendapatkan informasi pada server Raspberry Pi harus berada dalam satu jaringan. Alamat IP perangkat yang akan mengakses harus berada pada kisaran (*range*) dengan alamat IP Raspberry Pi. Alamat IP yang digunakan pada Raspberry Pi adalah 10.1.30.1/24. Artinya alamat 10.1.30.1 dengan *subnet* 24 atau 255.255.255.0 Jaringan tersebut mencakup 252 perangkat yang dapat terhubung dengan jaringan server. Sehingga tanpa mendapat alamat IP dari udchpd, mengakses sistem server secara manual dapat mengonfigurasi sendiri alamat IP perangkat secara statis. Selain menggunakan laptop, percobaan dilakukan dengan menggunakan perangkat ponsel (telepon seluler). Nama SSID yang muncul adalah Monitoring.



Gambar 2.32 Perangkat ponsel terhubung server

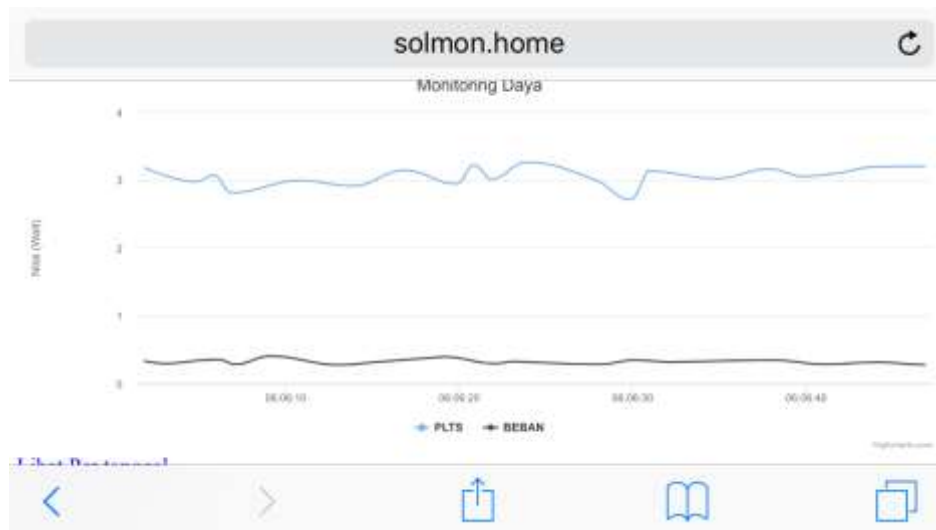
Ketika ponsel dinyalakan wi-fi-nya dan mencoba untuk memasuki jaringan server, alamat IP sudah terkonfigurasi secara otomatis. Selain dapat mengakses hasil keluaran sistem yang berupa grafik, juga dapat mengakses pengaturan *database* melalui PHPmyAdmin. Mengakses PHPmyAdmin cukup dengan menambahkan nama PHPmyAdmin pada nama domain yaitu <http://solmon.home/phpmyadmin>.

Mengatur sistem atau melihat *database* dengan memanfaatkan PHPmyAdmin memerlukan *username* root beserta *password*-nya. Tampilan sistem PHPmyAdmin pada ponsel memiliki fitur yang sama dengan tampilan sistem pada laptop.



Gambar 2.33 Alamat IP pada ponsel

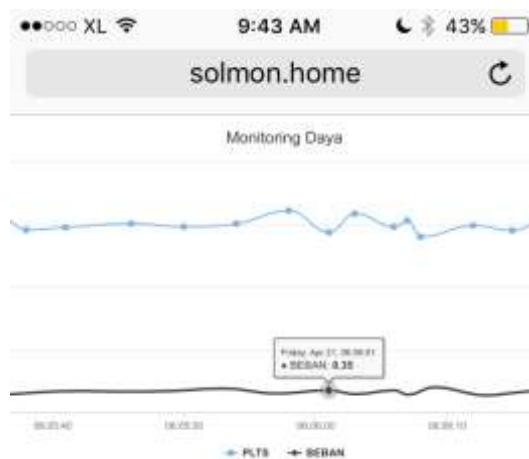
Setelah dipastikan terhubung, dengan menggunakan *browser* pada ponsel dapat mengakses halaman web yang ada pada server. Cukup memasukkan alamat domain <http://solmon.home> *browser* akan mengarahkan menuju grafik informasi pencatatan daya secara waktu nyata.



Gambar 2.34 Grafik pencatatan daya

Kelebihan dari tampilan informasi daya ini adalah data daya yang ditampilkan bersifat waktu nyata atau *real-time*, grafik yang mudah dibaca dan

dapat dilihat detailnya. Beberapa *node* dapat dilihat grafiknya secara bersamaan dengan menumpuknya jika berada pada nilai yang sama. Perbedaan antara grafik antar *node* adalah dengan membedakan warna grafiknya. Namun kelemahan dari tampilan grafik ini adalah ketika terjadi salah satu *node* tidak aktif maka grafik *node* lainnya akan berkerumun pada paling depan grafik dan grafik terlihat aneh. Kelemahan ini dapat diatasi dengan mengklik *node* yang tidak aktif pada bagian bawah *node*.



Gambar 2.35 Grafik pencatatan daya dengan informasi

Tampilan web pada ponsel sama sesuai dengan tampilan pada web di laptop meskipun menggunakan berkas yang sama di web server. Terdapat grafik dan pemilihan tanggal tersedia untuk melihat data daya yang disimpan pada *database*.



Gambar 2.36 Pemilihan tanggal pada ponsel

Tanggal yang terdapat pada halaman web ini merupakan kode javascript yang disimpan pada berkas `index.html`. Menampilkan data per tanggal ini memanfaatkan *query* pada MySQL dan operasi penjumlahannya juga merupakan fasilitas yang terdapat MySQL dalam bentuk *query*.

3.2 Pembahasan Hasil Penelitian

Perangkat Raspberry Pi 2 B dipilih dalam sistem pencatatan data produksi daya listrik pada sel surya karena perangkat tersebut memiliki bagian-bagian yang diperlukan dalam sistem yang dirancang ini, di antaranya memiliki *port* GPIO yang dapat digunakan oleh modul NRF24L01 sebagai modul komunikasi nirkabel dengan Arduino. Raspberry Pi 2 B ini bekerja layaknya sebuah komputer PC desktop namun dengan penggunaan daya yang rendah dibanding dengan komputer PC pada umumnya. Komunitas pengguna dari Raspberry Pi 2 B ini luas, dapat ditemukan di media sosial seperti Google+, Facebook, Twitter, dan juga forum komunitasnya tersendiri pada halaman web resminya. Sistem yang dirancang ini lebih terpusat konfigurasinya pada perangkat Raspberry. Selain

Regy Roy Hidayat Putra, 2018

RANCANG BANGUN SISTEM PENGAWASAN DAYA DAN ENERGI PADA PEMBANGKIT LISTRIK TENAGA SURYA MENGGUNAKAN RASPBERRY PI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

menggunakan Raspberry Pi, dapat menggunakan perangkat sejenis SBC lain seperti Banana Pi, Orange Pi, C.H.I.P. Namun kesulitan untuk mencari perangkat-perangkat tersebut tidak semudah Raspberry Pi yang dapat mudah ditemukan.

Rancangan sistem pencatatan produksi daya dan energi listrik oleh sel surya didukung oleh dua papan sirkuit yang bekerja, yaitu Raspberry dan Arduino. Arduino Nano yang bekerja sebagai *node* yang menggunakan modul NRF24L01 sehingga dapat berkomunikasi dengan Raspberry Pi. Meskipun dapat menggunakan jenis Arduino lain, namun Arduino Nano pertimbangan sebagai perangkat yang ideal untuk diterapkan pada sistem ini. Jika menggunakan Arduino UNO, dimensinya yang lebih besar dari Arduino Nano juga akan banyak *pin* yang tidak terpakai. Jika menggunakan Arduino lain yang lebih kecil dari Arduino Nano seperti Arduino Gemma maka sistem akan menjadi tidak efektif karena memiliki *pin* yang sangat sedikit.

Program-program penunjang untuk sistem yang terdiri dari 4 program bekerja secara terintegrasi. Pengaturan jam diperlukan ketika Raspberry Pi 2 B berada dalam keadaan tidak diberikan *input* daya, sehingga diperlukan pengaturan jam untuk menyinkronkan jam pada Raspberry Pi 2 B dengan waktu saat itu. Kelemahan lain pada sistem ini seperti kalibrasi sensor-sensor, seperti contoh pada kalibrasi sensor arus agar nilai yang dihasilkan terstandar dengan alat ukur lainnya. Pada kalibrasi sensor arus ini dapat dipengaruhi oleh *noise* karena bekerja tiga tahap yaitu mengubah nilai satuan arus yang diukur menjadi nilai magnetik dan kemudian diubah menjadi nilai analog 5V. Tentunya diperlukan logika yang kuat untuk memrogram pada Arduino agar nilai yang dibaca oleh sensor menjadi stabil.

Rancangan dari sistem yang telah dibuat memiliki keuntungan mobilitas dan kepraktisan pada pengguna yang akan mengakses informasi mengenai data daya dan energi listrik tanpa terkendala jarak seperti pada sistem yang menggunakan kabel. Penelitian yang dilakukan pada bangunan dengan luas bangunan 10x10 meter, terdapat dinding dengan ketebalan rata-rata 20cm menjadi rintangan dari sistem. *Node*, server, dan perangkat pengguna masih dapat bekerja dengan baik pada lokasi tersebut.

Daya yang dihasilkan oleh PLTS dipengaruhi oleh beberapa faktor di antaranya, nilai resistor atau nilai beban yang dialiri arus, intensitas cahaya, dan suhu. Daya puncak atau daya yang secara optimal dihasilkan oleh PLTS bergantung pada karakteristik solar panel. Sehingga perubahan beban dapat berpengaruh pada nilai daya yang dihasilkan.

Untuk pengembangannya, ketika program utama sebagai pencatat daya, Raspberry dapat mengoperasikan program lain pada satu waktu, sehingga merangkap menjadi media hiburan yang dapat diterapkan *home media center*. Selain itu, karena pada dasarnya perangkat pengguna yang terhubung melalui jaringan Wi-Fi dan memiliki nama domain layaknya halaman web pada internet maka sistem ini memiliki potensi untuk dikembangkan menjadi sistem yang terhubung dengan internet sehingga menjadi bagian dari *Internet of Things*.