



UNIVERSITAS INDONESIA

**RANCANG BANGUN DISTRO LINUX MONITORING PADA
SISTEM MONITORING PHOTOVOLTAIC**

SKRIPSI

RAHMAT PURWOKO

0706199786

**FAKULTAS TEKNIK ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO**

DEPOK

JULI 2010



UNIVERSITAS INDONESIA

**RANCANG BANGUN DISTRO LINUX PADA
SISTEM MONITORING PHOTOVOLTAIC**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

RAHMAT PURWOKO

0706199786

**FAKULTAS TEKNIK
DEPARTEMEN TEKNIK ELEKTRO
UNIVERSITAS INDONESIA
DEPOK
JULI 2010**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini sebagai syarat kelengkapan mejadi Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Tenik Elektro, Univeristas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari mulai perkuliahan sampai dengan penyusunan skripsi ini, sangatlah sulit untuk menyelesaikannya. Oleh karena itu perkenannkan saya untuk menyampaikan terima kasih kami kepada

- (1) Bapak Fery Yusivar, selaku dosen pembimbing
- (2) Kedua orang tua saya yang tak henti-hentinya memberikan motivasi untuk menyelesaikan kuliah.
- (3) Adik-adik saya, yang telah banyak memberikan dukungan
- (4) Bapak kepala Lembaga Sandi Negara, Ketua STSN yang telah mengijinkan saya untuk menempuh tugas belajar di jenjang S1 Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia.
- (5) Rekan- rekan di lingkungan STSN dan Pusdiklat Bumi Sanapati, Bogor
- (6) Sahabat yang telah banyak membantu dalam perkuliahan dan penyusunan skripsi

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini dapat membawa manfaat bagi ilmu pengetahuan.

Depok, 28 Juni 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini:

Nama : Rahmat Purwoko

NPM : 0706199786

Program Studi : Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif (Non-Exclusive Royalty-Free Right)** atas karya saya yang berjudul :

**RANCANG BANGUN DISTRO LINUX SERVER PADA SISTEM
MONITORING PHOTOVOLTAIC**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan , mengelola dalam bentuk pangkalan data (*database*), merawat dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 28 Juni 2010

Yang menyatakan



(.....)

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Rahmat Purwoko
NPM : 0706199786
Program Studi : Sarjana Ekstensi Teknik Elektro
Judul Skripsi : Rancang Bangun Distro Linux Monitoring pada
Sistem Monitoring Photovoltaic

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. Ir. Feri Yusivar, M.Eng (.....)

Penguji : Dr. Ir. Abdul Muis, ST, M.Eng (.....)

Penguji : Muhammad Salman, ST, M.IT (.....)

Ditetapkan di :

Tanggal :

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar

Nama : Rahmat Purwoko

NPM : 0706199786

Tanda Tangan : 

Tanggal : 28 Juni 2010

Universitas Indonesia

ABSTRAK

Nama : Rahmat Purwoko
Program Studi : Teknik Elektro
Judul : Rancang Bangun Distro Linux Server Monitoring pada Sistem Monitoring photovoltaic

Photovoltaic merupakan salah satu Renewable Energy yang mulai diteliti pemanfaatannya. Sistem monitoring untuk photovoltaic di Universitas Indonesia secara realtime sudah diteliti sebelumnya. System tersebut dibuat dengan platform system operasi komersial dan system komunikasi yang digunakan adalah system serial (RS232). Agar monitoring dapat diakses dari jaringan baik local maupun internet, maka perlu dibuat aplikasi sistem monitoring yang berbasis web. Rancang bangun Distro Linux server monitoring ini dimaksudkan untuk mempermudah monitoring pada photovoltaic. Aplikasi monitoring berbasis web yang dirancang menggunakan software-software Open Source serta dikemas dalam suatu distro Linux server. Dengan lisensi GPL (GNU Public License), diharapkan distro ini dapat digunakan pada instalasi sistem photovoltaic tanpa harus menggunakan banyak biaya.

Kata kunci:

Open Source, linux, server, monitoring, photovoltaic, rancang bangun, web

ABSTRACT

Name : Rahmat Purwoko
Program : Electro
Title : Design Server Monitoring Linux Distro for Photovoltaic Monitoring System

Photovoltaic is one of the renewable energy start investigated utilization. University of Indonesia have investigated photovoltaic systems for realtime monitoring. The system was created with a commercial operating system platform with used communication is a serial communication (RS232).

This thesis investigated the design of Linux distributions that have been installed server applications, web-based monitoring system. The research is intended to create a monitoring system using free software and open source and packaged in an open source server linux distro that can be applied on the photovoltaic system with a cheaper cost.

The results of this paper is a linux distro monitoring server with a web-based frontend to monitor the photovoltaic system so that the results of monitoring to be undertaken can be accessed through both local and Internet networks.

keyword: *Open Source, linux, server, monitoring, photovoltaic, design, web*

DAFTAR GAMBAR

Gambar 2.1.	UDP message format.....	14
Gambar 2.2.	Konfigurasi pinATmega32.....	16
Gambar 2.3.	Block Diagram NM7010A.....	18
Gambar 2.4.	Blok Diagram I2Chip W3100A.....	19
Gambar 2.5.	Modul DCS-01.....	20
Gambar 2.6.	Penggunaan DCS-01 untuk pengukuran beban.....	20
Gambar 2.7.	Grafik keluaran module DCS-01 terhadap arus yang disensor.....	21
Gambar 3.1.	Diagram blok server monitoring.....	22
Gambar 3.2.	Diagram Alur Server.....	25
Gambar 3.3.	Diagram Alur Agent	27
Gambar 3.4.	Diagram Alur MySQL Connection.....	29
Gambar 3.5.	Tampilan Login.....	29
Gambar 3.6.	Flowchart Login.....	30
Gambar 3.7.	Tampilan Grafik.....	31
Gambar 3.8.	Tampilan Backup.....	32
Gambar 3.9.	Diagram Testboard.....	33
Gambar 3.10.	Diagram alur Firmware.....	36
Gambar 3.10.	Diagram alur distro server	40
Gambar 4.1.	Implementasi pada Jaringan LAN.....	41
Gambar 4.2.	Server Monitoring.....	42
Gambar 4.3.	Testboard.....	43
Gambar 4.4.	Wireless AP I.....	44
Gambar 4.5.	Wireless AP II.....	44
Gambar 4.6.	Hasil capture wireshark.....	48
Gambar 4.7.	Hasil grafik multimeter	48
Gambar 4.8.	Hasil grafik monitoring.....	49
Gambar 4.9.	Capture Wireshark dengan data UDP lebih dari 255 Bytes.....	51

Gambar 4.10.	Pola grafik photovoltaic menggunakan Multimeter.....	53
Gambar 4.11.	Pola grafik photovoltaic menggunakan Server Monitoring.....	54



DAFTAR TABEL

Tabel 3.1.	Spesifikasi computer host	38
Tabel 4.1.	Konfigurasi Server Monitoring.....	44
Tabel 4.2.	Spesifikasi computer Server Monitoring	44
Tabel 4.3.	Konfigurasi Testboard	45
Tabel 4.4.	Konfigurasi Wireless AP 1.....	45
Table 4.5.	Konfigurasi Wireless AP 2.....	46
Tabel 4.6.	Output Multimeter Fluke dan testboard	49

BAB V

PENUTUP

5.1. KESIMPULAN

Dari hasil implemetasi dan pengujian didapatkan hasil sebagai berikut :

1. Proses Pembuatan CD Server Monitoring terdiri dari dua tahap yaitu :
 - a. Persiapan
 - b. Proses
2. Distro Server Monitoring dapat diimplementasikan pada jaringan Local Area Network (LAN)
3. Hasil karakteristik dari grafik yang didapatkan dari alat ukur dan server monitoring menunjukkan adanya kesamaan karakteristik. Adanya perberdaan yang terlihat dari setiap titiknya dikarenakan adanya perbedaan tingkat ketelitian dari sensor pada testboard dan alat ukur.
4. Dari hasil pengujian yang dilakukan batas minimal delay adalah 1 detik karena untuk melakukan pengambilan data ADC memerlukan waktu agar hasil dari ADC stabil. Selain itu pada proses pemaketan data dengan protocol UDP juga memerlukan waktu.
5. Untuk data maksimal yang dapat dikirimkan oleh Testboard adalah 255 bytes, hal ini merupakan batasan yang ditentukan pada Testboard, karena data tidak dibagi dalam beberapa kali pengiriman akan tetapi dalam satu kali pengiriman saja, sehingga data lebih dari 255 bytes akan dipotong dan sisanya didrop/dibuang.

SARAN

Beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut sebagai berikut:

- a. Optimalisasi kernel Linux untuk system realtime
- b. Penambahan dukungan protokol lain, seperti TCP, modbus dan lain sebagainya.
- c. Optimalisasi plugins untuk berbagai format system monitoring

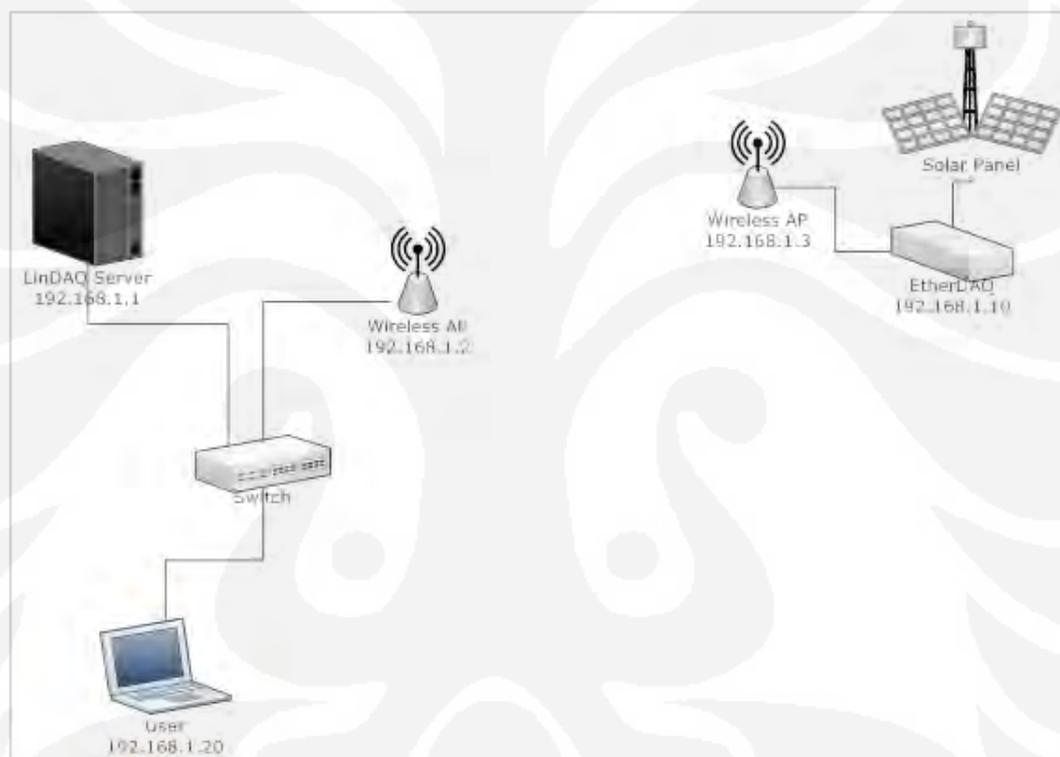
- d. Penambahan notifikasi kepada system Administrator
- e. Optimalisasi source code aplikasi system monitoring agar dapat menampilkan informasi yang lebih baik



BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. IMPLEMENTASI

Pada implementasinya, aplikasi ini dapat digunakan untuk jaringan local (LAN) maupun menggunakan internet seperti MAN dan WAN. Berikut ini gambaran skenario implementasi aplikasi monitoring pada jaringan LAN.



Gambar 4.1. Implementasi pada Jaringan LAN

Sumber : data olahan

Konfigurasi jaringan LAN yang digunakan pada pengujian system monitoring photovoltaic sebagai berikut :

- a. Solar Panel dan Testboard berlokasi di Gedung Engineering Center (EC), Fakultas Teknik Universitas Indonesia.
- b. Server berlokasi di Gedung Departemen Teknik Elektro Universitas Indonesia.

- c. Wireless Access, sebagai penghubung gedung EC dan Gedung Departemen Teknik Elektro.

Tabel-tabel dibawah ini merupakan konfigurasi jaringan untuk tiap-tiap node.

Tabel 4.1. Konfigurasi Server Monitoring

Server Monitoring	
Address	192.168.1.100
Subnet Mask	255.255.255.0
Gateway	192.168.1.1

Tabel 4.2. Spesifikasi komputer Server Monitoring

Server Monitoring	
Processor	Xeon, 2.6 GHz
Harddisk	500 GB
Memory	4 GB
Arsitektur	64 bit



Gambar 4.2. Server Monitoring

Sumber : data olahan

Tabel 4.3. Konfigurasi Testboard

Testboard	
Address	192.168.1.9
Subnet Mask	255.255.255.0
Gateway	192.168.1.1

Sumber : data olahan



Gambar 4.3. Testboard

Sumber : data olahan

Tabel 4.4. Konfigurasi Wireless AP 1

Wireless AP 1 (EnGenious Senao 2610)	
Address	192.168.1.2
Subnet Mask	255.255.255.0
Gateway	192.168.1.1

Sumber : data olahan

Tabel 4.5. Konfigurasi Wireless AP 2

Wireless AP 2 (EnGenious Senao 2610)	
Address	192.168.1.11
Subnet Mask	255.255.255.0
Gateway	192.168.1.1



Gambar 4.4. Wireless AP I

Sumber : data olahan



Gambar 4.5. Wireless AP II

Sumber : data olahan

4.1.1. Langkah-langkah Pengujian

Pengujian lebih difokuskan pada sensor tegangan photovoltaic. Dengan konfigurasi open maka arus yang diperoleh menjadi sangat kecil. Konfigurasi untuk pengujian dengan instalasi peralatan sebagai berikut :

- a. Photovoltaic, Photovoltaic yang digunakan untuk pengujian ini adalah photovoltaic yang sudah terinstalasi di gedung EC, Universitas Indonesia.
- b. Testboard, dihubungkan dengan photovoltaic.
- c. Server, instalasi server dilakukan di Gedung Departemen Teknik Elektro Universitas Indonesia
- d. Wireless Access Point, wireless access point diperlukan untuk meghubungkan gedung EC dan gedung Departemen Teknik Elektro
- e. Multimeter sebagai alat ukur standar pembanding digunakan Fluke 189 multimeter digital dengan tingkat akurasi $\pm 0.025\%$ dan mempunyai fungsi logging menggunakan data serial (RS232).

Pengujian dilakukan pada hari Selasa 7 Juli 2010 pada jam 12.30 – 16.30 dengan kondisi cuaca cerah.

4.1.2. Pengujian I

Pengujian ini dilakukan untuk mengetahui apakah distro yang dibuat sudah dapat digunakan dengan cara instalasi pada computer server. Tahapan – tahapan yang dilakukan sebagai berikut :

- a. Persiapan CD distro server
- b. Instalasi komputer server
- c. Konfigurasi Server, Agent, plugins dan Web Frontend
- d. Pengambilan data dan pengiriman data melalui Testboard.

Distro Server dirancang untuk menajankan proses instalasi secara otomatis sehingga tinggal memasukan CD distro server dan instalasi akan berjalan secara otomatis.

Konfigurasi yang dilakukan adalah konfigurasi network. Secara default konfigurasi network distro server sebagai berikut :

```
Address      : 192.168.1.1
Netmask     : 255.255.255.0
Gateway     : 192.168.1.1
```

Untuk dapat disesuaikan dengan konfigurasi untuk pengujian dilakukan konfigurasi ulang dengan cara :

a. Login

```
$ login : root
$ password : root
```

b. Edit file network interfaces

```
$ nano /etc/network/interfaces
auto eth0
iface eth0 inet static
address      192.168.1.100
netmask     : 255.255.255.0
gateway     : 192.168.1.1
```

c. Restart network

```
$ /etc/init.d/networking restart
```

Langkah selanjutnya adalah konfigurasi Server, Agent dan plugins dengan menjalankan perintah.

```
$ rems-configure
```

Lanjutkan langkah-langkah disesuaikan dengan konfigurasi pengujian. Pada distro server ini secara default konfigurasi untuk Server, Agent dan plugins sudah sesuai, sehingga konfirmasi yang diperlukan hanyalah persetujuan dari konfigurasi Server, Agent dan plugins.

Untuk menyakinkan apakah server dan Agent berjalan dengan baik dapat dilakukan dengan menjalankan perintah :

```
$ ps aux | grep rems
```

Apabila terlihat hasil sebagai berikut berarti server dan Agent sudah berjalan sebagai service/daemon. Sebaliknya jika tidak ada maka perlu dijalankan sebagai berikut :

```
$ /etc/init.d/rems-server
```

```
$ /usr/sbin/rems-server {start|stop}
$ /etc/init.d/rems-server start
    REMS Server starting
$ /etc/init.d/rems-agent start
```

Langkah selanjutnya adalah menghubungkan Testboard dengan jaringan dan mulai melakukan pengiriman data. Untuk dapat melihat log yang masuk dapat digunakan perintah :

```
$ ls -l /var/log/daq/
$ tail -f /var/log/192.168.1.10.log
```

Untuk dapat melihat data dan grafik, dilakukan dengan membuka browser dan menunjuk ip address server. Langkah ini merupakan langkah yang harus dikerjakan agar dapat melihat data dan log dari grafik.

- a. Login sebagai admin
- b. Create data
- c. Create grafik
- d. Monitoring.

4.1.3. Pengujian II

Pengujian ini dimaksudkan untuk mendapatkan perbandingan antara hasil pengukuran menggunakan multimeter dan pengukuran menggunakan tesboard. Konfigurasi peralatan dapat dilihat seperti pada gambar 4.

Dengan menggunakan wireshark dilakukan capture data jaringan, untuk melihat pengiriman data yang dilakukan Testboard ke server monitoring. Untuk lebih detil dilakukan filtering dengan ip address source 192.168.1.9 yaitu ip address dari Testboard. Hasil capture sebagai berikut :

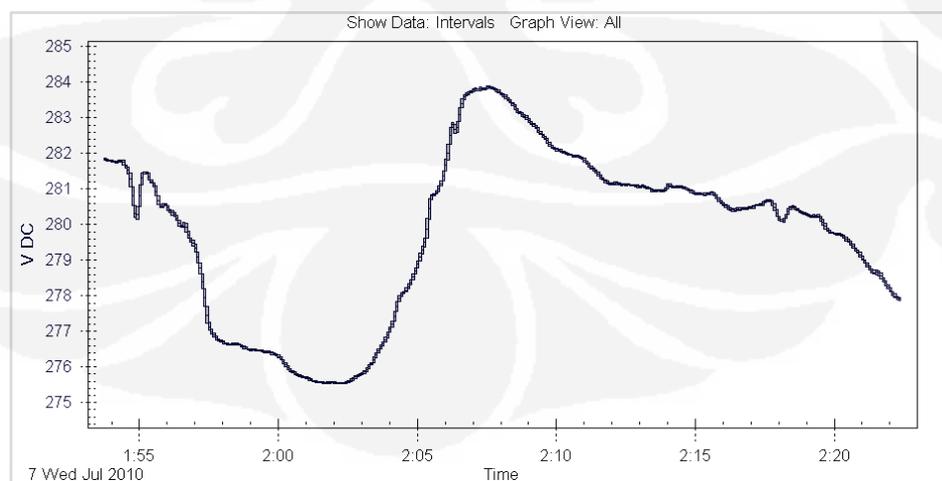
No. -	Time	Source	Destination	Pr
556	153.313223	192.168.1.9	192.168.1.105	UT
557	153.313259	192.168.1.9	192.168.1.105	UT
563	160.333417	192.168.1.9	192.168.1.105	UT
564	160.333457	192.168.1.9	192.168.1.105	UT
571	168.788320	192.168.1.9	192.168.1.105	UT
572	168.788356	192.168.1.9	192.168.1.105	UT

0000	00 15 b7 0e e1 5c 0c 80	0c 22 38 4e 08 00 45 00\.. "8N..E.
0010	01 1b 00 02 40 00 40 11	b6 0d c0 a8 01 09 c0 a8@.@.
0020	01 69 27 0f 27 0f 01 07	7b 84 31 39 32 2e 31 36	.i.....{.192.16
0030	38 2e 31 2e 39 20 64 61	74 61 3a 20 32 37 36 2e	8.1.9 da ta: 276.
0040	31 39 35 3b 30 2e 30 39	38 3b 00 00 00 00 00 00	195;0.09 8;.....
0050	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00c0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00f0	00 00 00 00 ff 00 00 00	00 00 00 00 00 00 00 00
0100	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0110	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0120	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

Gambar 4.6. Hasil Capture Wireshark

Sumber : data olahan

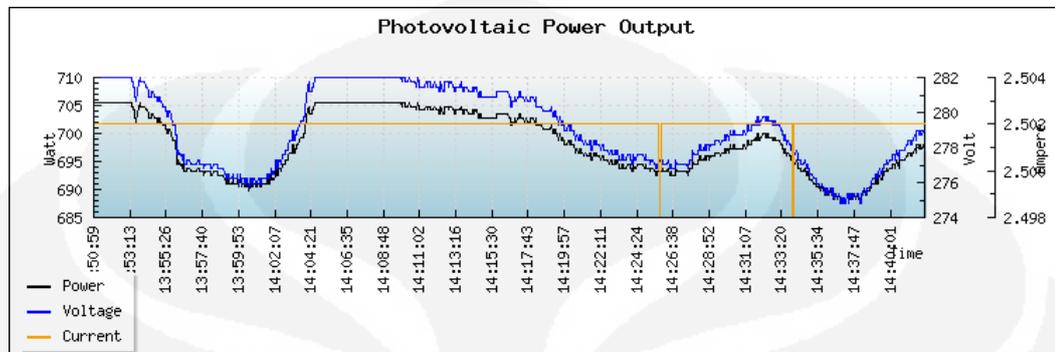
Pengujian dilakukan dengan menghubungkan secara parallel output dari photovoltaic pada multimeter dan sensor tegangan pada selama kurang lebih 30 menit dengan rata-rata interval pengambilan data kurang lebih 5 detik. Hasil pengukuran melalui multimeter didapatkan grafik yang terlibat pada gambar 4.



Gambar 4.7. Hasil Grafik multimeter Fluke

Sumber : data olahan

Hasil log yang terukur dari sensor pada Testboard, kemudian dikirimkan ke server dan ditampilkan melalui web. Grafik yang diperoleh pada pengukuran yang bersamaan dengan pengukuran multimeter terlihat seperti gambar 4.



Gambar 4.8. Hasil grafik monitoring

Sumber : data olahan

4.1.4. Pengujian III

Percobaan yang dilakukan adalah dengan cara mengubah-ubah delay waktu antara pengiriman data-data dari ADC. Percobaan pertama dilakukan dengan selang waktu pengiriman 8 detik. Pada percobaan ini didapatkan hasil normal seperti yang terlihat pada gambar 4. Didapatkan log dari logfile sebagai berikut :

```

2010-07-06 16:10:44 192.168.1.9 data: 262.979;2.502
2010-07-06 16:10:52 192.168.1.9 data: 262.979;2.502
2010-07-06 16:10:53 192.168.1.9 data: 262.979;2.502
2010-07-06 16:11:01 192.168.1.9 data: 262.979;2.502
2010-07-06 16:11:01 192.168.1.9 data: 262.979;2.502
2010-07-06 16:11:09 192.168.1.9 data: 262.979;2.502
2010-07-06 16:11:10 192.168.1.9 data: 262.979;2.502
2010-07-06 16:11:18 192.168.1.9 data: 262.979;2.502
2010-07-06 16:11:26 192.168.1.9 data: 262.979;2.502
2010-07-06 16:11:27 192.168.1.9 data: 263.255;2.502

```

Percobaan kedua dilakukan dengan menggunakan selang waktu pengiriman 1 detik. Dari percobaan didapatkan hasil log sebagai berikut :

```

2010-07-06 16:02:30 192.168.1.9 data: 270.974;2.502

```



```

200138 1407.385012      192.168.1.9      192.168.1.100    UD
200139 1407.830700      192.168.1.9      192.168.1.100    UD
200140 1407.830726      192.168.1.9      192.168.1.100    UD

```

```

⊞ Frame 200139 (297 bytes on wire, 297 bytes captured)
⊞ Ethernet II, Src: 0c:80:0c:22:38:4e (0c:80:0c:22:38:4e), Dst: Vmware_22:e4:a6 (
⊞ Internet Protocol, Src: 192.168.1.9 (192.168.1.9), Dst: 192.168.1.100 (192.168.
⊞ User Datagram Protocol, Src Port: distinct (9999), Dst Port: distinct (9999)
    Source port: distinct (9999)
    Destination port: distinct (9999)
    Length: 263
    ⊞ Checksum: 0xbf15 [validation disabled]
⊞ Data (255 bytes)
    Data: 3139322E3136382E312E3920646174613A203237302E3639...
    [Length: 255]

```

```

0000  00 0c 29 22 e4 a6 0c 80 0c 22 38 4e 08 00 45 00  ...)".... ."8N..E.
0010  01 1b 02 25 40 00 40 11 b3 ef c0 a8 01 09 c0 a8  ...%@.@. ....
0020  01 64 27 0f 27 0f 01 07 bf 15 31 39 32 2e 31 36  .d'.'.... ..192.16
0030  38 2e 31 2e 39 20 64 61 74 61 3a 20 32 37 30 2e  8.1.9 da ta: 270.
0040  36 39 38 3b 32 2e 35 30 32 61 61 61 61 61 61 61  698;2.50 2aaaaaaa
0050  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
0060  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
0070  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
0080  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
0090  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
00a0  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
00b0  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
00c0  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
00d0  61 61 61 61 61 61 61 61 61 61 61 61 61 61 61  aaaaaaaaa aaaaaaaa
00e0  61 61 61 61 61 61 61 61 61 61 61 61 62 62 62  aaaaaaaa aaaaabbb
00f0  62 62 62 62 62 62 62 62 62 62 62 62 62 62 62  bbbbbbbb bbbbbbbb
0100  62 62 62 62 62 62 62 62 62 62 62 62 62 62 62  bbbbbbbb bbbbbbbb
0110  62 62 62 62 62 62 62 62 62 62 62 62 62 62 62  bbbbbbbb bbbbbbbb
0120  62 62 62 62 62 62 62 62 62 62 62 62 62 62 62  bbbbbbbb b

```

Gambar 4.9. Capture Wireshark dengan data UDP lebih dari 255 bytes

Sumber : data olahan

4.2. ANALISA

Dari data yang diperoleh, maka Testboard sudah berjalan dengan baik. Terjadi pengiriman paket-paket UDP yang berisi data-data dari Testboard. Hal ini dapat dilihat dari hasil capture dari jaringan LAN dan display grafik yang menunjukkan data tegangan dan arus dari photovoltaic. Capture data dari jaringan menggunakan software wireshark.

Pada pengujian pertama secara angka yang diperoleh terdapat perbedaan antara hasil yang diperoleh dari multimelter dan sensor tegangan. Secara tabular

hasil selama kurang lebih 1 menit dari pengambilan data tersebut dapat dilihat sebagai berikut :

Tabel 4.6. Output Fluke Multimeter dan Testboard

Waktu	Fluke 189 Multimeter	Testboard	Error
2010-07-07 02:00:03	276.2359	276.487	0.091%
2010-07-07 02:00:08	276.1589	276.211	0.019%
2010-07-07 02:00:13	276.0638	276.211	0.053%
2010-07-07 02:00:18	275.9703	275.935	0.013%
2010-07-07 02:00:23	275.9072	276.211	0.110%
2010-07-07 02:00:28	275.8624	275.66	0.073%
2010-07-07 02:00:33	275.8124	275.935	0.044%
2010-07-07 02:00:38	275.7838	276.211	0.155%
2010-07-07 02:00:43	275.7605	275.935	0.063%
2010-07-07 02:00:48	275.7232	276.211	0.177%
2010-07-07 02:00:53	275.7108	276.211	0.181%
Jumlah	3034.989	3037.218	0.980331
Rata-rata	275.9081	276.1107	0.0891%

Sumber : data olahan

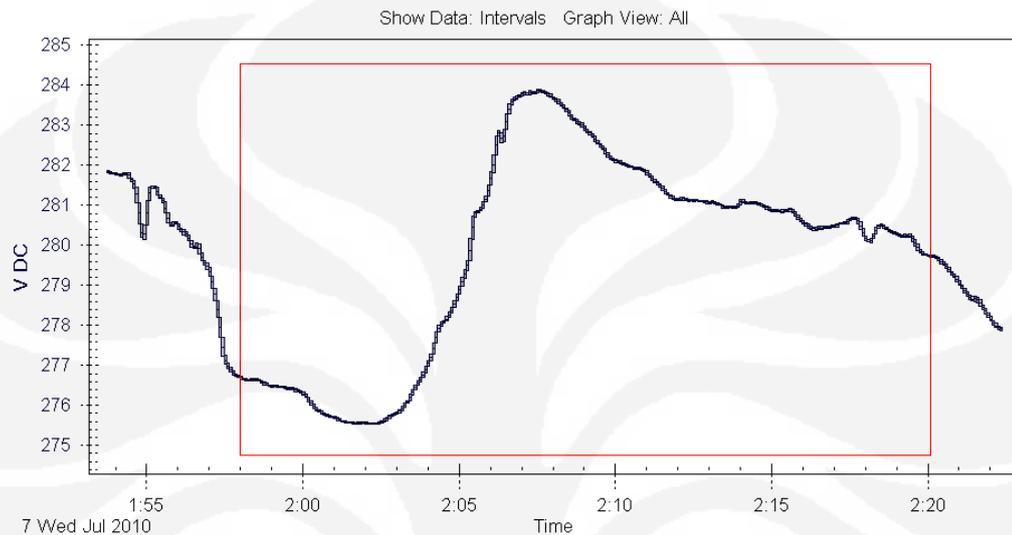
Persentase error dihitung menggunakan data hasil multimeter fluke 189 sebagai acuan, sehingga rumus dihitung sebagai berikut :

$$\left| \frac{\text{Hasil Output Multimeter} - \text{Hasil Output Testboard}}{\text{Hasil Output Multimeter}} \right| \times 100\%$$

Sehingga selisih antara hasil pengukuran multimeter dengan pengukuran dari sensor tegangan pada testboard yang didapatkan sebagai berikut :

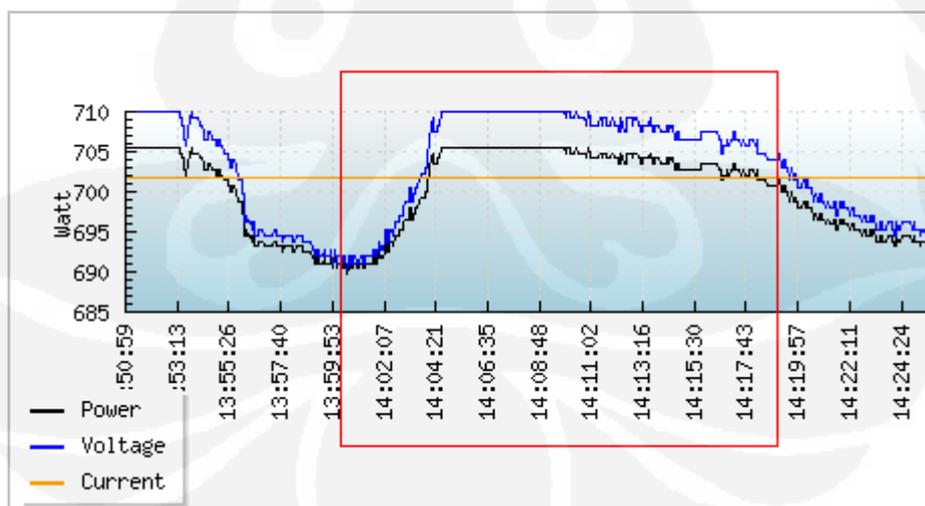
$$\begin{aligned} \Delta V &= (\Sigma \text{Hasil Output Multimeter} - \Sigma \text{Hasil Output testboard})/n \text{ data} \\ &= (3034.989-3037.218)/11 \\ &= -0.20262 \end{aligned}$$

Namun demikian dari grafik yang didapat terlihat mempunyai pola yang hampir sama antara grafik dari hasil multimeter dengan grafik dari sensor tegangan pada saat yang relatif sama. Disebut dengan relative sama karena terdapat besar selisih pengambilan data dalam orde millisecond.



Gambar 4.10 Pola grafik photovoltaic menggunakan multimeter

Sumber : data olahan



Gambar 4.11. Pola grafik photovoltaic pada Server Monitoring

Sumber : data olahan

Pada pengujian selanjutnya, dari hasil log yang diperoleh menunjukkan bahwa pengambilan data dengan selang waktu 5 detik masih memungkinkan untuk mendapatkan data log yang baik dalam arti masih terbaca.

Percobaan selanjutnya yaitu percobaan dengan selang waktu 1 detik masih dapat menampilkan data seperti seharusnya. Dari log yang diperoleh terlihat beberapa perulangan yang terjadi.

Pada pengujian III, yaitu percobaan dengan pengiriman data lebih dari 255 bytes data dan selang waktu pengiriman ± 8 detik didapatkan hasil bahwa data yang dikirimkan melalui Ethernet hanya 255 bytes saja dan data sisanya tidak dikirimkan pada paket berikutnya. Hal ini menunjukkan bahwa data yang dikirimkan sesuai dengan konfigurasi dasar pengirimannya yaitu 255 bytes, selebihnya akan di drop/dibuang.

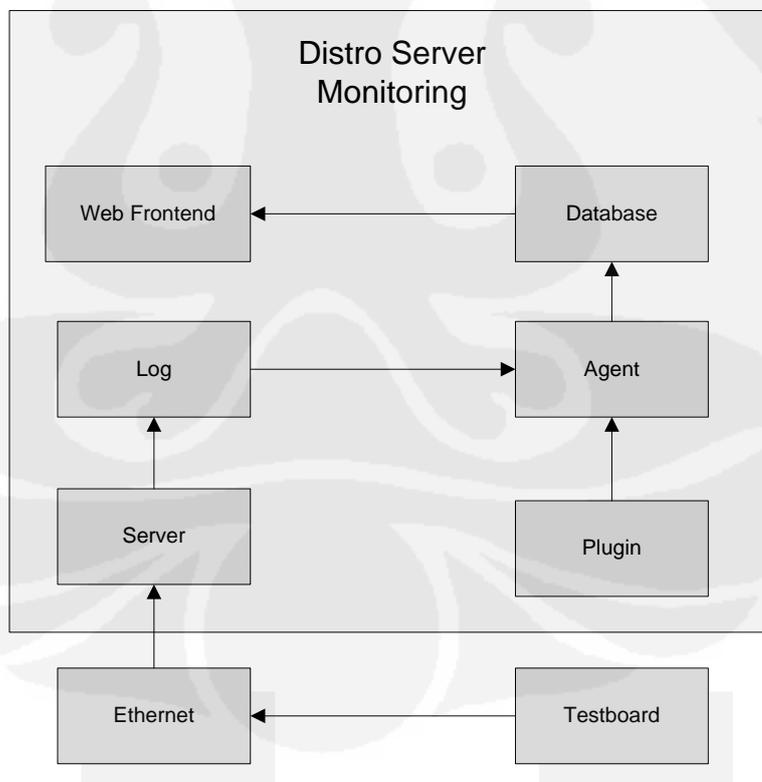
BAB III PERANCANGAN DISTRO MONITORING

3.1. SOFTWARE

Software Monitoring berbasis web yang dirancang terdiri dari beberapa komponen sebagai berikut :

- a. Server, program daemon yang memonitor soket UDP dan menulis logfile.
- b. Agent, program daemon yang memonitor log, parsing dan insert data ke database MySQL
- c. Frontend Web, frontend software monitoring berbasis web
- d. Database, MySQL Database sebagai penyimpanan record data.

Secara umum arsitektur software sistem monitoring dapat dilihat pada gambar 3.1.



Gambar 3.1. Diagram blok Server Monitoring

Sumber : data olahan

3.1.1. Server dan Agent

Server & Agent Monitoring dirancang sebagai service/daemon. Ada beberapa alasan yang mendasari hal tersebut yaitu:

- a. Server melakukan monitoring socket UDP dan pengambilan data secara terus menerus.
- b. Agent memonitor log data dari Server dan melakukan parsing serta inserting data ke database.

Untuk dapat menjadikan skrip Server dan Agent menjadi daemon dan service dibutuhkan proses pembuatan daemon dengan cara forking terhadap proses OS. Python mempunyai suatu module yang berhubungan dengan OS, yaitu module os.

Untuk dapat menggunakan module dalam python cukup sederhana yaitu dengan cara mengimport module tersebut “import <nama module>”. Dalam skrip dapat dilakukan sebagai berikut :

```
def daemonize(self):
    try:
        pid = os.fork()
    except OSError, e:
        sys.exit(1)
    if (pid == 0): # The first child.
        os.setsid()
        try:
            pid = os.fork() # Fork a second child.
        except OSError, e:
            raise Exception, "%s [%d]" % (e.strerror,
            e.errno)
        sys.exit(1)
    if (pid == 0): # The second child.
        os.chdir("/")
        os.umask(0)
    else:
        open(self.pidfile, 'w').write("%d" % pid)
        logger.info("process on %d" % pid)
        os._exit(0)
else:
```

```
os._exit(0)
```

Setelah proses daemon dijalankan kemudian dilakukan proses dilanjutkan dengan membuka port dan menunggu datagram yang dikirimkan oleh client. Karena UDP merupakan *connectionless* protocol, maka tidak diperlukan proses koneksi. Proses yang dilakukan adalah membuat socket dan mengidentifikasi socket tersebut sebagai datagram.

Komponen yang diperlukan dalam proses membuat socket adalah address dan port. Proses ini melibatkan socket dalam hal ini AF_INET sedangkan tipe socket datagram dideklarasikan dengan SOCK_DGRAM. Sebaliknya untuk tipe TCP menggunakan SOCK_STREAM.

Address dikonfigurasi ke "0.0.0.0" yang berarti dapat menerima dari semua IP address. Berikut ini fungsi untuk membuka socket dan mengambil data dari datagram dan selanjutnya disimpan pada file log.

```
def open(self):
    logger.info("Attempting %s" % self.host + " to open socket
on port (%s)" % self.port + " (%s) " % self.size)
    try:
        if not os.path.isdir(self.daqdir):
            os.mkdir(self.daqdir)
            logger.info("create daq directory")
            logger.info("directory found!")
    except:
        logger.info("Log directory error!")
        sys.exit(1)
    try:
        self.udpsoc=socket.socket(socket.AF_INET, \
socket.SOCK_DGRAM)
        self.udpsoc.bind((self.host, self.port))
        logger.info("Socket opened, start receiving data")
        while 1:
            logger.info("try to receive data ...")
            data,addr=self.udpsoc.recvfrom (self.size)
            if not data:
                logging.info ("not data")
```

```

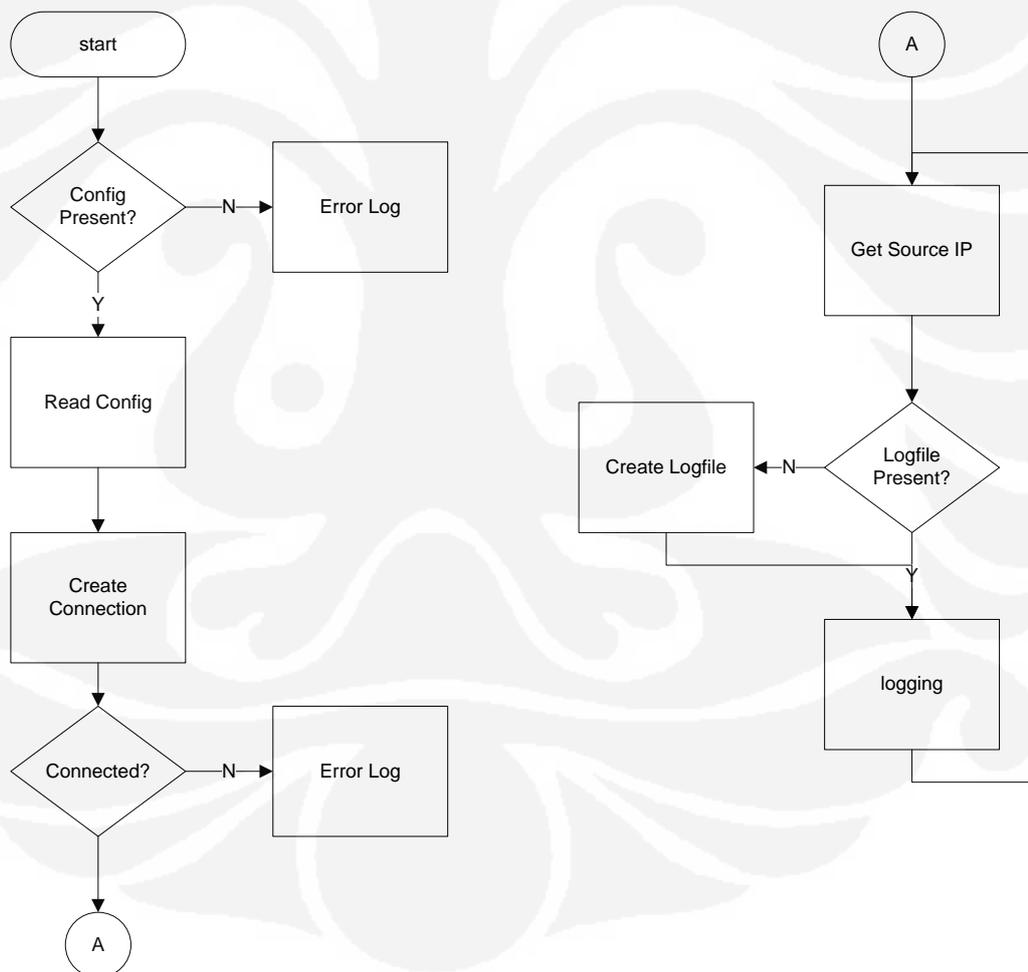
else:
    logs = addr[0]+".log"
    self.file=open(os.path.join(self.daqdir,
logs), 'a')

    self.file.write(data)
    if (data[-1] != "\n"):
        self.file.write("\n")
    self.file.close()

except:
    logger.info("failed")

```

Secara umum, proses server dapat digambarkan pada diagram alur pada gambar 3.2 dibawah ini :



Gambar 3.2. Diagram alur Server

Sumber : data olahan

Perbedaan antara Server dan Agent adalah server lebih kepada pengambilan data dari datagram sedangkan pada Agent, setelah proses daemon terjadi kemudian Agent melakukan proses koneksi ke database MySQL dan kemudian melakukan proses parsing log dengan terlebih dahulu membaca file-file konfigurasi dan log.

Untuk melakukan koneksi ke database MySQL digunakan modul Python-mysqldb. Penggunaan sebagai berikut :

```
con = MySQLdb.connect(host = self.dbhost, user = self.dbuser, \
    passwd = self.dbpass, db = self.dbname)
self.cursor = con.cursor()
```

Proses membuka log menggunakan suatu mekanisme *tailFollow* yang mendeteksi adanya perubahan file log kemudian melakukan pengambilan baris dari file log dan melakukan parsing *pattern* (pola) teks yang disimpan pada file konfigurasi plugins. Apabila pattern match, maka log akan di parsing menjadi query sql yang kemudian diinsert ke dalam database

Proses parsing data dari log menggunakan module python re (*Regular Expression*). Langkah-langkah yang parsing secara umum adalah :

- a. Mengambil pola regex pada file konfigurasi
- b. Melakukan kompilasi regex
- c. Mengambil string yang akan diparsing
- d. Melakukan *pattern match* dengan regex
- e. Apabila sesuai maka akan didapatkan kumpulan grup dari string yang sesuai dengan pola dari regex

Pola-pola regex disimpan dala suatu file konfigurasi yang bernama plugins. Selain regex, pada plugins disimpan data-data mengenai client seperti address dan nama file log. Untuk setiap client yang mengirima mempunyai satu file konfigurasi. Pada *pseudocode* berikut ini dapat dilihat proses parsing yang dilakukan pada Agent.

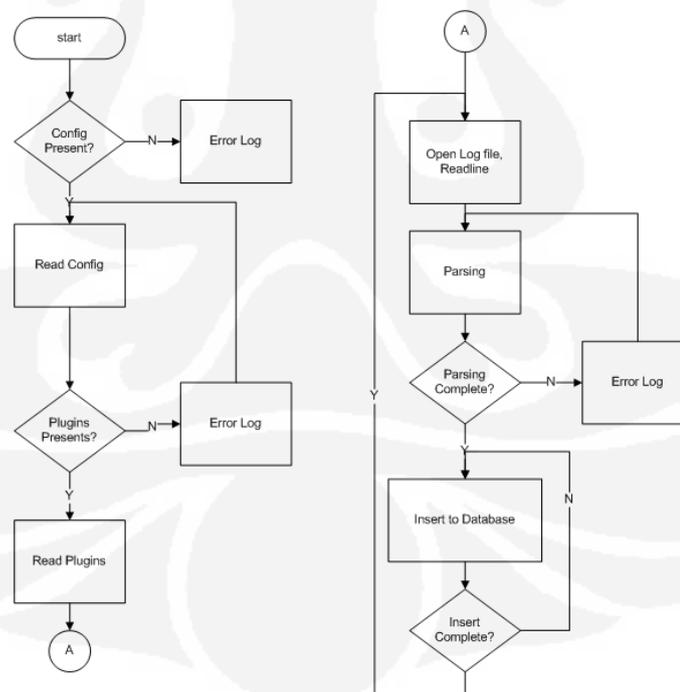
```
def process(self):
    m = re.compile(self.regexp)
```

```

tail = TailFollow(self.logfile, track=1)
while 1:
for line in tail:
    if not line:
        time.sleep(1)
    else:
        ml = m.match(line)
        a = ml.groupdict()
        strs = ''
        str_insert = ", ".join(map(str, a.keys()))
        str_values = "', '".join(map(str, a.values()))
        self.query = "INSERT INTO data("+str_insert+") \
VALUES('"+str_values+"')"
        self.cursor.execute(self.query)
        time.sleep(1)
tail.close()

```

Secara umum, gambaran proses yang dilakukan oleh Agent dapat dilihat pada diagram alur gambar 3.3 dibawah ini.



Gambar 3.3 Diagram alur Agent

Sumber : data olahan

3.1.2. Frontend

Frontend merupakan Software berbasis web yang dibuat menggunakan bahasa pemrograman PHP. Frontend ini berjalan pada web server apache. Secara garis besar frontend terdiri dari beberapa komponen :

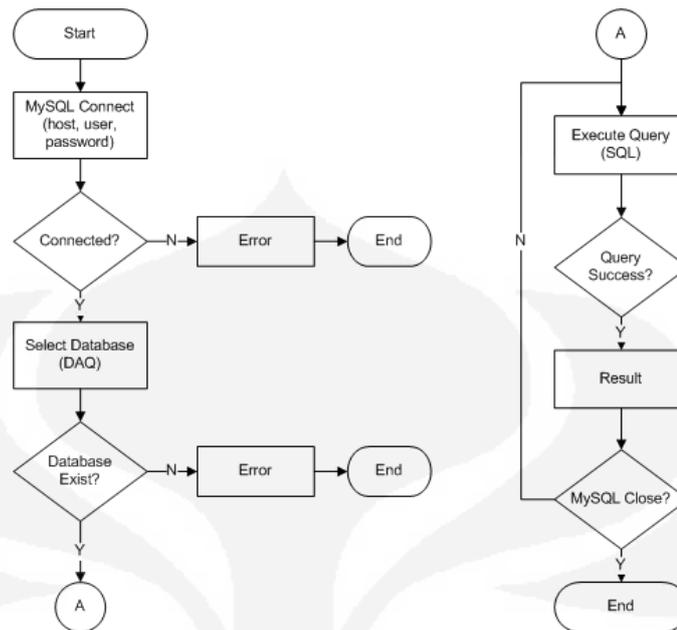
- a. Database Connection
- b. User Management
- c. Graph & Data Manajemen
- d. Utility

Untuk dapat berhubungan dengan mysql database aplikasi frontend diperlukan parameter-parameter koneksi MySQL seperti host, port, username, login dan databse yang akan digunakan.

PHP5 sudah menyediakan *library* untuk dapat berhubungan dengan MySQL, namun pada aplikasi ini dibuat suatu class khusus untuk koneksi ke database MySQL.

Class PHP ini diberi nama db.php, yang secara umum berfungsi sebagai jembatan antara aplikasi frontend dengan database seperti untuk dapat mengambil data, menghapus, maupun menyimpan record dalam database. Secara umum diagram alurnya sebagai berikut :

- a. Membuat koneksi dengan mengirimkan parameter koneksi seperti host, username dan password.
- b. Memilih Database yang akan digunakan dalam hal ini database tersebut adalah DAQ.
- c. Mengeksekusi perintal SQL, perintah-perintah SQL yang akan dieksekusi merupakan perintah-perintah SQL dari aplikasi frontend ang digenerate sesuai dengan kebutuhan aplikasi.
- d. Menutup koneksi, setelah selesai maka koneksi ke database MySQL dapat ditutup.



Gambar 3.4 Diagram Alur MySQL Connection

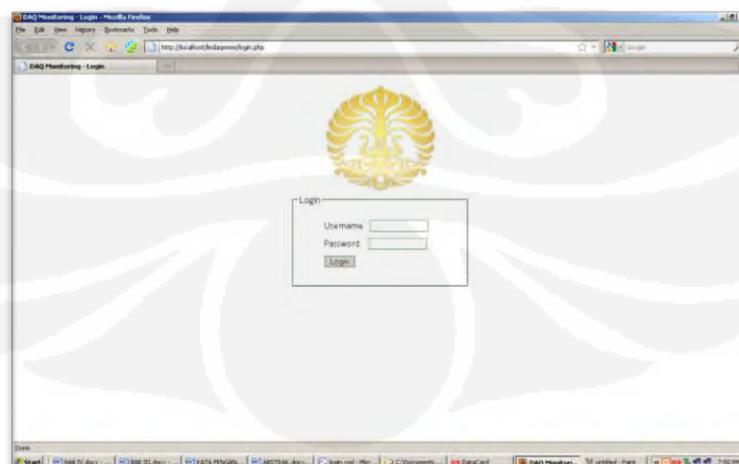
Sumber : data olahan

User group aplikasi frontend dirancang menjadi dua group yaitu :

a. Admin

Mempunyai kewenangan untuk membuat, menghapus user. Selain itu admin dapat menambah aset (hardware monitoring) dan membuat grafik dari data log.

b. User, merupakan user yang hanya dapat melihat halaman-halaman yang disetujui oleh admin

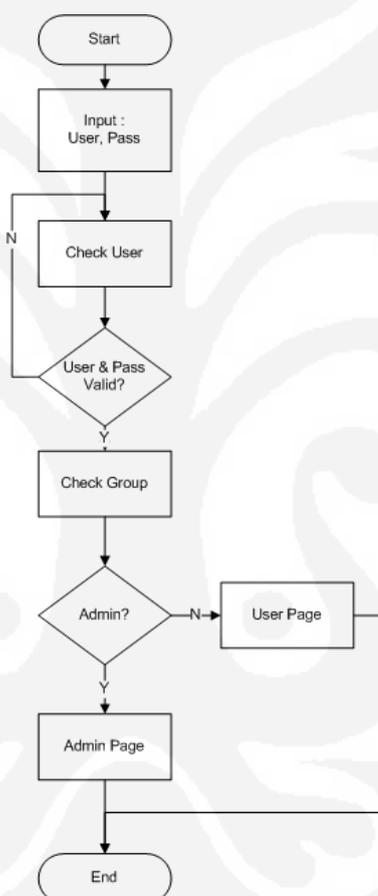


Gambar 3.5 Tampilan Login

Sumber : data olahan

User dalam aplikasi frontend ini hanya dapat melihat data/grafik yang disetujui oleh admin. Informasi admin dan user tersimpan dalam database oleh karena itu saat login, admin atau user harus mengisi username dan password yang valid (terdaftar).

Tampilan login berisi dua parameter yaitu username dan password. Gambar tampilan login sebagai berikut :



Gambar 3.6. Flowchart Login

Sumber : data olahan

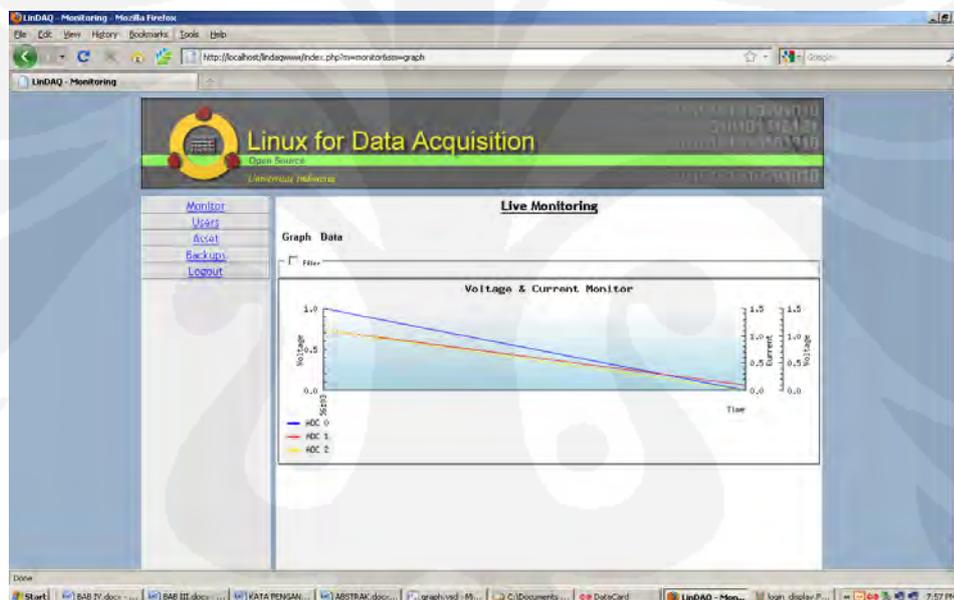
Grafik merupakan representasi dari data monitoring. Library yang digunakan untuk grafik adalah JpGraph. JpGraph merupakan aplikasi dual *license*, salah satu licensenya adalah free. Untuk dapat menggunakan JpgGraph, pada script PHP diincludekan library JpGraph.

Tipe grafik yang dirancang dalam aplikasi frontend web adalah tipe plot, dengan jumlah plot yang ditampilkan menyesuaikan dengan konfigurasi dari admin yang membuat grafik.

Sama halnya dengan penampil grafik, banyaknya jumlah kolom data log disesuaikan dengan konfigurasi saat pembuatan grafik tersebut. Secara default agrafik dan data menampilkan semua field dari hasil parser dari data log system monitoring.

Pada penampil graph ditambahkan filtering untuk melihat secara detil dari plot-plot hardware monitoring yang mengirimkan log ke server.

Berikut rancangan tampilan graph.



Gambar 3.7 Tampilan Grafik

Sumber : data olahan

Pada software Frontend ini dirancang utility tambahan untuk mendukung frontend agar lebih aplikatif. Utility tersebut yaitu :

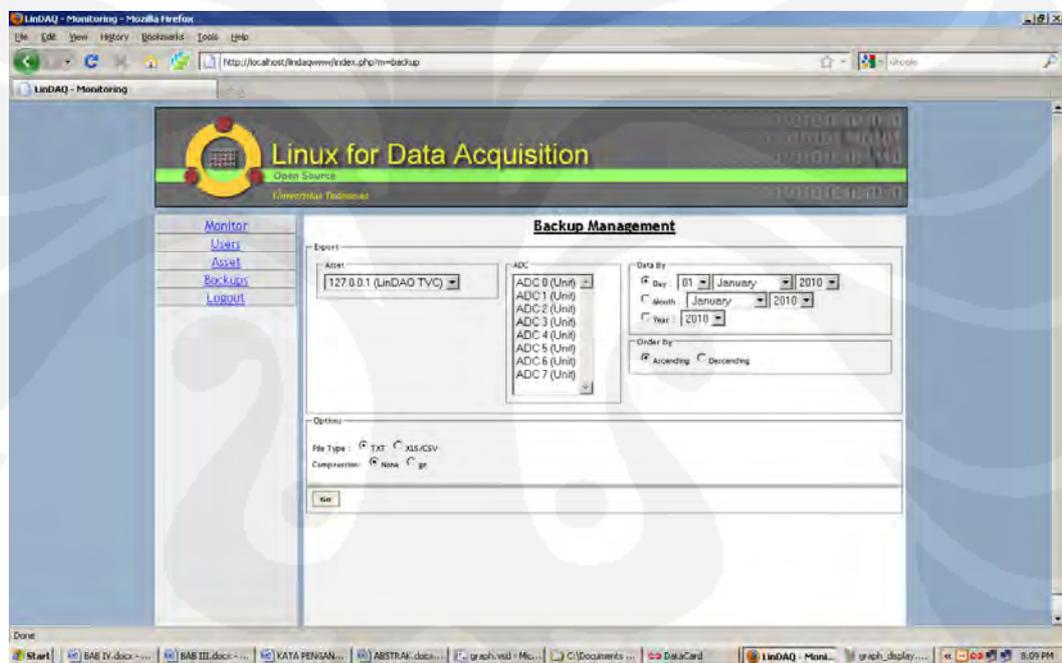
a. Search

Search merupakan utility untuk pencarian record data. Parameter yang digunakan adalah range waktu antara pada data yang dicari.

b. Download

Utility download berfungsi untuk membackup data log. backup yang dirancang adalah partial backup, yang artinya setelah pencarian dilakukan, record dapat diunduh oleh user.

Format dasar dari utility download adalah text file, dengan sedikit penambahan header dan separator, user dapat mengunduh file dalam format csv (comma separated values), yang dapat dibuka menggunakan Microsoft Excell. Selain itu pula disediakan failitas compress untuk kompresi file yang diunduh.



Gambar 3.8 Tampilan Backup

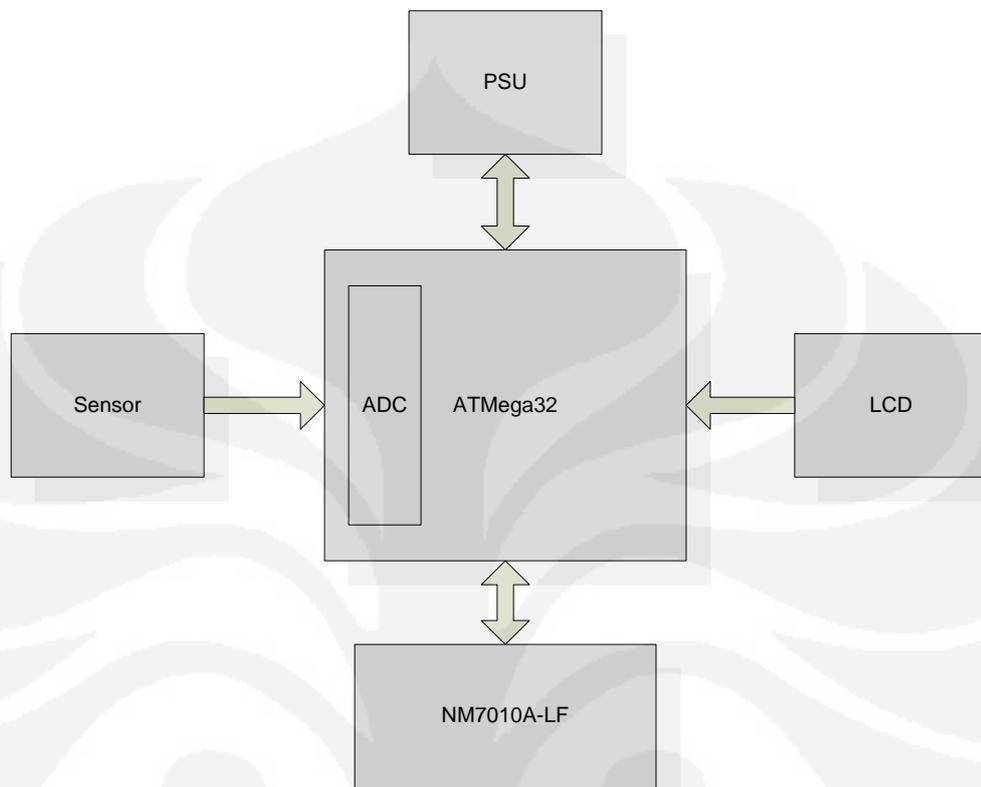
Sumber : data olahan

3.2. HARDWARE

3.2.1. Firmware

Hardware dirancang menggunakan microcontroller ATmega32 dan penambahan interface lainnya seperti : komunikasi RS232, RTC, EEPROM dan Network module NM7010A LF, yang mempunyai interface I2C. pada network modul inilah proses pemaketan data yang akan dikirimkan menggunakan protocol

UDP berlangsung. Diagram blok yang menunjukkan hubungan antara ATmega32 dan interface lain dapat dilihat pada gambar 3.9 Dibawah ini.



Gambar 3.9 Diagram Blok Testboard

Sumber : data olahan

Testboard memiliki konfigurasi sebagai berikut:

- Power Supply Unit
- ADC
- LCD Display
- Komunikasi ethernet
- Sensing unit

Power supply yang digunakan pada testboard untuk memenuhi kebutuhan microcontroller ATmega 32 dan bagian-bagian lainnya. Tegangan digunakan 5 V DC dan 12 V DC. Pada penulisan ini digunakan adaptor CDROM external yang telah memiliki tegangan tersebut. Alasan penggunaan adaptor external ini karena tegangan yang dibutuhkan terpenuhi dan Arus yang dibutuhkan hanya 1.5 A.

Bagian Sensing terdiri dari dua buah sensor yaitu sensor tegangan dan sensor arus. Sensor tegangan memonitor tegangan output dari photovoltaic. Konfigurasi awal digunakan pembagi tegangan dengan tambahan buffer dan op-amp dengan konfigurasi non-inverting.

Sensor tegangan berfungsi untuk menurunkan tegangan output photovoltaic dari 300V menjadi 5 V sehingga ADC internal dari ATmega 32 dapat membacanya. Secara perhitungan, kurang lebih penurunan tersebut 60 kali. Sensor arus yang digunakan adalah DCS-01 yang dihubungkan dengan beban.

Secara umum testboard ini mengikuti konfigurasi dari penelitian sebelumnya, yaitu “Sistem Monitoring photovoltaic secara Realtime” (Elsa, 2009), dengan penambahan RTC, EEPROM dan module Ethernet.

3.2.2. Firmware

Firmware dibuat dengan menggunakan compiler BASCOM versi 1.1.9. Setelah firmware dicompile, maka firmware didownload ke IC ATmega32.

Flowchat dari firmware

- a. Inisialisasi
- b. Pembacaan ADC dari Microcontroller
- c. Konversi
- d. Display LCD
- e. Pengiriman data lewat UDP

Proses inisialisasi untuk ADC dalam pemrograman BASCOM AVR Konfigurasi ADC yang digunakan adalah single dengan prescaler auto dan reference ADC yang digunakan AVCC (2.54 V). Sehingga menggunakan dapat dilihat *pseudocode* sebagai berikut :

```
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
```

ADC yang digunakan adalah 10 bit ADC, dan channel yang digunakan adalah 2 buah yaitu channel 1 untuk sensor voltase dan channel 2 untuk sensor

arus. Isi dari channel 1 dan channel 2 ADC ATmega32 akan dikirimkan melalui protokol UDP pada bagian message.

Sebelum dapat mengirimkan data dengan UDP maka harus dideklarasikan library yang akan digunakan, dalam hal ini library `tcpip.lbx` yang merupakan library untuk chip network W3100A, chip yang digunakan pada network module NM7010A-LF. Konfigurasi dasar berupa konstanta dan yang diperlukan dalam pemaketan data menunjuk pada alamat-alamat register pada W3100A. Dengan mode yang digunakan I2C, maka diperlukan dua buah koneksi ke bus I2C ATmega32, yaitu SDA dan SCK,

Selain itu dibutuhkan koneksi untuk interrupt pada saat inisialisasi dan sebelum pengiriman data. Karena ADC dan NM7010A-LF menggunakan interrupt maka pada code BASCOM diaktifkan (enable) interrupt ini dihubungkan dengan INT0. Setelah interrupt enable maka baru bisa diaktifkan konfigurasi dasar network pada NM7010 seperti pemberian nama address, netmask, gateway dan port. Secara umum diagram alur untuk firmware dapat dilihat pada gambar 3.4:

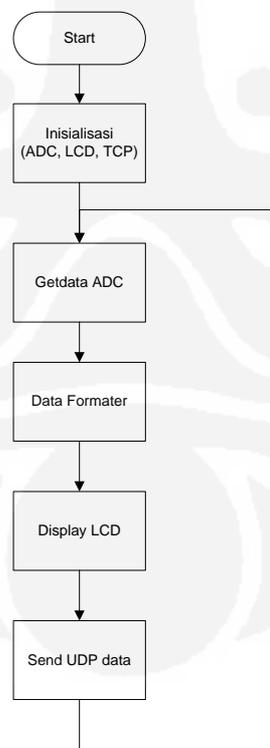
```
$lib "lcd4busy.lib"
'constants
Const Sock_stream = $01
Const Sock_dgram = $02
Const Sock_ip1_raw = $03
Const Sock_mac1_raw = $04
Const Sel_control = 0
Const Sel_send = 1
Const Sel_recv = 2

'socket status
Const Sock_closed = $00
Const Sock_arp = $01
Const Sock_listen = $02
Const Sock_symsent = $03
Const Sock_symsent_ack = $04
Const Sock_synrecv = $05
Const Sock_established = $06
Const Sock_close_wait = $07
```

```
Const Sock_last_ack = $08
Const Sock_fin_wait1 = $09
Const Sock_fin_wait2 = $0a
Const Sock_closing = $0b
Const Sock_time_wait = $0c
Const Sock_reset = $0d
Const Sock_init = $0e
Const Sock_udp = $0f
Const Sock_raw = $10

Config Pind.4 = Output
Set Portd.4
Reset Portd.4
Waitms 12
Set Portd.4

Enable Interrupts
Config Tcpi = Int0 , Mac = 12.128.12.34.56.78 , Ip = 192.168.1.9
, Submask = 255.255.255.0 , Gateway = 192.168.1.1 , Localport =
1000 , Tx = $55 , Rx = $55 , Twi = &HCC , Clock = 30000
```



Gambar 3.10 Diagram alur Firmware

Sumber : data olahan

3.3. DISTRO SERVER MONITORING

3.3.1. Persiapan

Pada tahap ini dilakukan persiapan untuk membuat distro server monitoring. Persiapan yang dibutuhkan adalah computer host dengan spesifikasi sebagai terlihat pada table 3. :

Tabel 3.1 Spesifikasi komputer host

Prosesor	Intel Centrino duo 1.6 GHz
Memory	1 GB
Hardisk	40 GB
OS	Ubuntu 8.0.4.4 LTS

Sumber : data olahan

Selanjutnya diinstall paket-paket software untuk membuat distro. Paket-paket tersebut adalah :

- a. UCK (Ubuntu Customization Kit), merupakan kumpulan skrip yang digunakan dalam kustomisasi ubuntu
- b. mksisos, merupakan software yang digunakan untuk packing iso distro
- c. Dh-make, merupakan software yang digunakan untuk membuat debian package
- d. Gnupg, merupakan software yang digunakan untuk membuat public key untuk digital signature paket dan iso image

Cara instalasi paket software dengan menggunakan perintah pada console sebagai berikut :

```
$ sudo apt-get install uck mksisos dh-make gnupg
```

Proses berikutnya adalah membuat software monitoring menjadi paket yang siap diinstalasi. Untuk memudahkan proses pembuatan, software monitoring dipecah menjadi tiga bagian yaitu :

- a. Library, merupakan kumpulan modul python untuk software monitoring
- b. Web, merupakan kumpulan skrip-skrip php (web)
- c. Apps, merupakan kumpulan skrip-skrip executable dan file-file konfigurasi.

Penamaan pada paket debian menggunakan menggunakan konvensi sebagai berikut :

<nama paket>-<versi>.

Versi merupakan grup angka yang menunjukkan versi dari paket. Sebagai contoh software monitoring ini mempunyai versi 0.1 (initial release) sehingga penamaan sebagai berikut :

- a. Library : rems-lib-0.1
- b. Web : rems-web-0.1
- c. Apps : rems-app-0.1

Masing masing dibuat folder tersendiri :

```
$ mkdir -p rems-0.1 rems-web-0.1 rems-apps-0.1
```

```
$ cd rems-app-0.1
```

```
$ cp /home/user/skrip usr/
```

```
$ dh_make -n
```

Edit file-file pada folder debian

Minimal yang harus diedit adalah :

- a. Control
- b. Changelog
- c. Compat
- d. Rules

Setelah file-file tersebut diubah, and disesuaikan dengan konfigurasi, kemudian build paket tersebut :

```
$ debuild -us -uc
```

Setelah itu maka akan didapatkan file package dengan nama `rems-apps_0.1_i386.deb`, `rems-lib_0.1.i386.deb` dan `rems-web_0.1.i386.deb`. File-file paket debian ini nantinya akan ditambahkan pada saat kustomisasi distro.

3.3.2. Proses pembuatan distro

Pembuatan distro server merupakan proses yang cukup mudah akan tetapi untuk menghasilkan distro yang sesuai dengan keinginan memerlukan proses yang berulang-ulang.

Langkah-langkanya adalah sebagai berikut :

- a. Download iso server ubuntu 8.0.4 LTS
- b. Buat folder tmp pada directory home

```
$ mkdir /home/user/tmp
```
- c. Proses unpacking ISO

```
$ sudo uck-remaster-unpack-iso
```

Pada proses ini maka file-file hasil ekstrasi dari iso file akan ditempatkan pada folder `tmp/remaster-iso`

- d. Proses prepare alternate

```
$ sudo uck-remaster-prepare-alternate
```

Pada proses ini akan ditambahkan folder extras pada `/tmp/pool`. Pada folder inilah paket-paket tambahan akan dimasukkan.

```
$ sudo cp -r rems-lib-0.1.i386.deb rems-app-0.1.i386.deb  
rems-app-0.1.i386.deb /tmp/pool/extras/rems
```

- e. Kustomisasi

Kustomisasi dilakukan pada file-file dibawah ini :

1. `Preseed.seed`, merupakan file yang mengatur proses instalasi linux. Pada file ini dilakukan kustomisasi untuk instalasi monitoring.
2. `Isolinux.cfg`, merupakan file konfigurasi yang menunjukkan file preseed yang akan di-load.

- f. Clean up
- g. Finalize alternate

```
$ sudo uck-remaster-finalize-alternate <gpgkey>
```

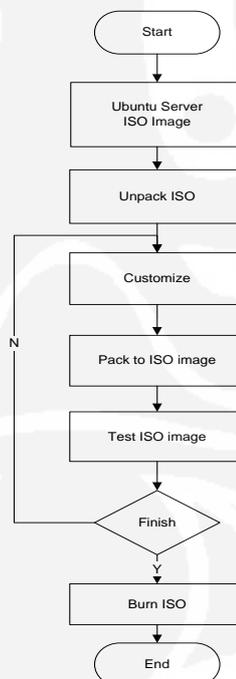
Pada proses ini membutuhkan gpgkey dari pembuat distro yang dapat degenerate dengan menggunakan gnupg.

```
$ gpg --gen-key
```

- h. Packing iso file

```
$ sudo uck-remaster-pack-iso
```

Proses packing untuk menjadi iso file. Hasil iso yang jadi ada pada directory /home/user/tmp/remaster-new-file/livecd.iso. Kemudian ubah menjadi livecd.iso menjadi rems-server_0.1.i386.iso, selanjutnya siap ditest menggunakan VMware maupun langsung diburn ke CD dan diinstall pada computer.



Gambar 3.11 Diagram alur pembuatan distro server

Sumber : data olahan

BAB II

LANDASAN TEORI

2.1. SISTEM OPERASI LINUX

Linux adalah sebuah sistem operasi yang sangat mirip dengan sistem UNIX [1]. Seluruh kode sumber linux termasuk *kernel*, *device driver*, *libraries* (perpustakaan sistem), program dan *tool* pengembangan disebarakan secara bebas dengan lisensi GPL (General Public License).

Sistem operasi Linux terdiri dari tiga bagian kode penting [1]:

- a. Kernel, bertanggung jawab memelihara semua abstraksi penting dari sistem operasi, termasuk hal-hal seperti memory *virtual* dan proses-proses.
- b. Perpustakaan sistem, menentukan kumpulan fungsi standar dimana aplikasi dapat berinteraksi dengan kernel, dan mengimplementasikan hampir semua fungsi sistem operasi yang tidak memerlukan hak akses penuh atas kernel.
- c. Utilitas sistem merupakan program yang melakukan pekerjaan manajemen secara individual

2.1.1. Distribusi Linux

Distro adalah kernel linux ditambah dengan kumpulan paket-paket perangkat lunak dari GNU dan lainnya, yang dibundel menjadi satu, dengan tujuan untuk mempermudah proses distribusi perangkat lunak[2]

Sebelum membuat distro perlu dipahami dua karakter distro yang dibuat, berdasarkan sifat redistribusinya yaitu[2]:

- a. Distro dipakai untuk untuk diri sendiri.

Distro ini dibuat dengan basis LFS (*Linux From Scratch*) dan semua aplikasi dikompilasi dalam *pristine code* (kode program murni).

- b. Distro dari turunan distro besar yang sudah maju.

Biasanya yang dipakai rujukan adalah Redhat (misal Fedora), debian (misal ubuntu).

2.1.2. Distro Ubuntu

Ubuntu adalah salah satu distribusi Linux berbasis pada Debian dan memiliki antarmuka desktop. Proyek Ubuntu disponsori oleh Canonical Ltd (perusahaan milik Mark Shuttleworth). Nama Ubuntu diambil dari sebuah konsep ideologi di Afrika selatan[3].

Ubuntu dikonsentrasikan untuk penggunaan, bebas digunakan, secara reguler dirilis setiap 6 bulan dan mudah diinstalasi. Versi terbaru adalah versi 10.4 yang dirilis April 2010. Distro Linux Ubuntu mendukung berbagai arsitektur komputer seperti[3] :

- a. PC (Intel x86).
- b. PC 64-bit (AMD64).
- c. PowerPC (Apple iBook dan Powerbook, G4 dan G5).
- d. Sun UltraSPARC.
- e. T1 (Sun Fire T1000 dan T2000)

2.1.3. Proses Pembuatan Distro

Tahap ini merupakan proses pembuatan distro server. *Source code* software dan cara pembuatan distro server dapat dilihat pada lampiran, Berikut ini akan dijelaskan secara umum proses pembuatan distro server, pengintegrasian distro server dan termasuk kustomisasi yang dilakukan.

Proses pembuatan distro server secara umum terdiri dari dua tahap yaitu[2][4] :

1. Persiapan spesifikasi kebutuhan system

Pada tahap ini adalah mempersiapkan kebutuhan sistem untuk pembuatan distro server. Software-software yang diperlukan dalam proses pembuatan distro Linux Server Monitoring sebagai berikut :

- a. Ubuntu Server 8.04.4 LTS
- b. Apache2 Web Server
- c. PHP5
- d. MySQL Database Server
- e. Python-mysqldb, Python library untuk MySQL *binding*.
- f. Software system monitoring.

2. Proses Pembuatan distro

Secara umum, proses kerja untuk pembuatan distro sebagai berikut[2] :

- a. Membuat Direktory kerja.
- b. Mengunduh CD Image Ubuntu 8.04.
- c. Mendownload Paket-paket software yang diperlukan dan dependenciesnya.
- d. Mount image Ubuntu CD.
- e. Membuat copy dari image.
- f. Menulis ulang isolinux/isolinux.cfg.
- g. Membuat preseed/remseed.
- h. Memasukkan paket-paket software yang diperlukan
- i. Mengepak CD Image

Setelah proses selesai, untuk mengecek hasil instalasi dari iso file yang telah dibuat dapat menggunakan VMWare.

2.2. OPEN SOURCE SOFTWARE

2.2.1. Apache Web Server

Web server merupakan suatu server internet yang menggunakan protokol HTTP (Hypertext Transfer Protocol) untuk melayani semua proses pentransferan data[5]. Hingga saat ini webserver merupakan server yang dapat dikatakan sebagai tulang punggung bagi semua internet. Hal ini dikarenakan web server tidak hanya melayani data berbetuk teks saja akan tetapi juga dapat menampilkan gambar dalam bentuk 2D maupun 3D, suara dan lain sebagainya.

Apache web server yang dibuat oleh Apache Software Foundation(ASF). ASF sendiri merupakan suatu komunitas para pengembang software yang kemudian setiap software yang mereka produksi berada dibawah lisensi Apache Licence yang berdifat bebas dan kode sumber terbuka.

Apache web server merupakan salah satu dari sekian banyak web server yang ada di internet, akan tetapi sebagian besar server dengan lingkungan Unix/Linux menggunakannya, hal ini disebabkan oleh beberapa alasan yaitu[6]:

- Kecepatan yang lebih baik dibandingkan dengan webserver lainnya.
- Performance yang sangat baik.
- Dapat diperoleh dengan gratis (open source)

Apache sendiri merupakan program yang modular dan memiliki program layanan pendukung yang cukup banyak yang dapat digunakan oleh pengguna. Beberapa layanan tersebut adalah[6] :

- Kontrol Akses
- *Common Gateway Interface* (CGI)
- *Personal Home Page*
- *Server Side Include* (SSI)

Apache *binary* atau biasa disebut dengan `httpd` di lingkungan sistem operasi Unix/Linux dan `Apache.exe` di lingkungan sistem operasi Microsoft Windows pada umumnya berjalan secara *background*. Tanpa melihat sistem operasi dimana apache diinstall, `httpd/apache` secara umum terdiri dari empat subdirektori, yaitu[6]:

- `conf`
Conf merupakan subdirektori tempat file-file konfigurasi dari apache. Salah satu yang terpenting adalah `httpd.conf`. dimana pada file tersebut merupakan konfigurasi dari web server apache dan menyatakan URLs yang dilayani.
- `htdocs`
Subdirektori ini merupakan subdirektori dimana nantinya file-file HTML yang di minta oleh browser client. Direktori-direktori yang berada dibawahnya merupakan data-data umum yang nantinya akan disediakan sesuai dengan permintaan dari client browser.
- `logs`
Subdirectory logs berisi file-file log yang mencatat akses terhadap web server, selain itu juga mencatat error yang terjadi saat pengaksesan.

- cgi-bin

Subdirektori ini merupakan subdirectory dimana skrip-skrip CGI disimpan. Skrip-skrip tersebut dijalankan mewakili permintaan dari client karena client tidak dimungkinkan secara langsung berinteraksi dengan sistem operasi.

2.2.2. MySQL Database Server

MySQL merupakan Database Management System (DBMS) Open Source. Sebagai database server[7], MySQL melayani banyak user yang ingin mengakses ke database-database. Proyek Pengembangan MySQL berada dibawah lisensi GNU General Public License atau GPL. Selain berlisensi GPL MySQL juga mempunyai lisensi propeteri yang dimiliki dan disponsori oleh perusahaan Swedia MySQL AB, namun sekarang kepemilikan MySQL AB sudah berada dibawah Sun Microsystem yang merupakan Subsidiary Oracle Coproration.

Database terdiri dari koleksi data yang terorganisasi untuk satu atau beberapa tujuan[7]. Salah satu cara pengklasifikasian isi data dengan database adalah melalui tipe data, sebagai contoh bibliograpy, gambar, angka dan lain sebagainya. Pada dasarnya software atau program mengorgasisai data merujuk pada model database yang umum seperti model relasional (relational model), model hirarki (hiearki model) dan model jaringan (network model).

Database-database merupakan penampung yang berbasis software yang mana secara terstruktur mengoleksi dan menyimpan informasi sehingga user dapat mengambil (retrieve), menambah (add), mengupdate (update) dan menghapus informasi secara otomatis.

Dengan kata lain program database merupakan program yang didesain untuk user sehingga mereka dapat menambah maupun mengurangi informasi yang diperlukan. Struktur database sendiri merupakan sebuah table yang berisi informasi dan terdiri atas baris dan kolom.

Beberapa contoh database yang cukup dikenal luas seperti Oracle, Microsoft SQL Server, Microsoft Access, MySQL, PostgreSQL, Dbase, SQLite dan lain sebagainya[8]. Sebagian database tersebut merupakan database server yang bias melayani banyak user dan banyak database.

Database Management Systems (DBMS) merupakan database yang mengorganisasi penyimpanan data. DBMS dapat mengontrol pembuatan, maintenance dan penggunaan struktur database dan para penggunanya. DBMS sendiri dapat digolongkan menurut database model yang disupport, sebagaimana telah disebutkan sebelumnya.

Setiap database model cenderung membuat bahasa query untuk mengakses database. Salah satu bahasa query untuk database relasional adalah Structured Query Language (SQL). Bahasa SQL merupakan bahasa digunakan untuk manajemen data pada Relational Database Management System (RDBMS).

SQL didasarkan pada aljabar relasi yang mempunyai ruang lingkup pada data query dan update, pembuatan skema database dan akses control. Secara umum SQL dibagi menjadi bentuk query, yaitu[8] :

- Data Definition Language (DDL), adalah sebuah metode query SQL yang berguna mendefinisikan data pada sebuah database.
- Data manipulation language (DML), merupakan metode query SQL yang berguna untuk memanipulasi tabel maupun database.

2.2.3. PHP

Hypertext Pre Processor atau kemudian disebut sebagai PHP adalah bahasa pemrograman dalam pengembangan web dinamis, berupa skrip-skrip yang diletakkan dan di eksekusi di server (server-side)[9].

PHP versi 5 sudah mendukung Object Oriented Programming (OOP). Beberapa kelebihan PHP dari bahasa pemrograman lain[9]:

- Bahasa pemrograman PHP adalah bahasa skrip yang tidak melakukan kompilasi dalam penggunaannya.
- Web server yang mendukung PHP cukup banyak seperti Apache, IIS, lighthttpd, Ngix hingga Xitami.
- Mempunyai dukungan komunitas yang cukup banyak milis, forum dan lain sebagainya.
- Merupakan open source software yang dapat dijalankan semua platform sistem operasi.

PHP merupakan server-side embedded script language yang artinya, sintaks-sintaksnya dan perintah yang diberikan akan sepenuhnya dijalankan server, akan tetapi disertakan pada halaman HTML biasa[9]. Aplikasi-aplikasi yang dibangun pada umumnya akan memberikan hasil pada halaman web, akan tetapi prosesnya secara keseluruhan dijalankan di server.

2.2.4. Python

Python merupakan bahasa pemrograman dinamis yang mendukung pemrograman berorientasi obyek[10]. Bahasa pemrograman ini dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan diberbagai platform sistem operasi. Seperti halnya bahasa pemrograman lain python seringkali digunakan sebagai skrip dengan interpreter yang terintegrasi dengan sistem operasi. Saat ini kode python dapat digunakan dalam beberapa sistem operasi seperti :

- UNIX/Linux
- Microsoft Windows
- Mac OS X
- Amiga
- Palm
- Symbian

Bahasa Pemrograman Python didistribusikan dengan lisensi yang berbeda dari beberapa versi python yang dirilis. Python dapat digunakan untuk kepentingan aplikasi komersial, dan hal ini tidak bertentangan dengan definisi Open Source Software maupun General Public License (GPL).

Python dikembangkan secara modular, yang berarti perlu modul tambahan untuk bisa membuat aplikasi-aplikasi yang berhubungan dengan sistem lain. Modul-modul standar dari Python antara lain modul os dan sys, yaitu modul untuk berinteraksi dengan sistem operasi dimana Python diinstall.

Daemon/service atau dalam lingkungan sistem operasi UNIX/linux biasa disebut Daemon merupakan background process yang didesain untuk bekerja terus-menerus dan otomatis. Sebagai contoh adalah server Apache yang secara

terus menerus melayani permintaan web. Daemon pada lingkungan Linux dibuat menggunakan aturan-aturan sebagai berikut[11] :

- a. *Fork* pada proses utama
- b. Mengubah mode file
- c. Membuka log untuk proses tulis
- d. Membuat Session ID (SID) yang unik
- e. Mengubah directory kerja ke tempat yang lain
- f. Menutup file descriptor standar
- g. Masuk pada mode daemon yang sebenarnya

Python sudah mendukung Socket Programming. Dengan pemrograman socket, aplikasi seperti client-server dengan protocol-protokol TCP/IP dapat dibuat. Pemrograman socket adalah cara untuk menggunakan Komponen/API (Application Programming Interface) untuk membuat sebuah aplikasi. Elemen-elemen penting yang terdapat pada socket sebagai berikut[12] :

- Protokol, merupakan protokol yang digunakan. Protokol yang umumnya dipakai dalam socket programming adalah TCP (connection oriented) dan UDP (connectionless)
- Local IP, Address dari komputer lokal
- Local Port, port dari komputer lokal
- Remote IP, Address dari komputer remote
- Remote Port, port dari komputer remote

Aplikasi Socket Programming secara umumnya terdiri dari dua kategori pengiriman data[13]:

- Stream Socket, *Connection-oriented sockets* yang menggunakan Transmission Control Protocol(TCP) atau Stream Control Transmission Protocol (SCTP)
- Datagram Socket, *Connectionless* (TCP) yang menggunakan User Datagram Protocol (UDP).

Stream socket (TCP) mengharuskan adanya koneksi terlebih dahulu, kemudian mengirimkan paket-paket data secara berurutan, penerima dijamin mendapatkan paket-paket secara berurutan dari paket yang pertama sampai dengan paket yang terakhir. TCP dapat menangani paket yang hilang, rusak, terpecah maupun terduplikasi. lain halnya dengan stream socket, datagram socket menggunakan UDP yang merupakan *connectionless protocol*, tidak memerlukan koneksi terlebih dahulu sehingga paket data dapat hilang ataupun penerimaanya tidak berurutan.

Port merupakan koneksi data virtual yang digunakan aplikasi untuk bertukar data secara langsung. Dalam sebuah komputer, terdapat banyak port, nomor port yang valid mulai dari 1-65535.

Nomor-nomor port dikategorikan dalam 3 jenis, yaitu:[13]

- Well-known Port, merupakan port yang sudah banyak dikenal penggunaanya. Port tersebut mulai dari nomor 0-1023.
- Registered Port, merupakan port yang sudah di registrasi oleh Internet Corporation for Assigned Name and Number(IANA). Port yang teregister pada range mulai dar 1024-49151
- Dynamic/Private Port, range port mulai dari 49152-65535

2.2.5. UDP

User Datagram Protocol (UDP) merupakan suatu protocol lapisan transfer TCP/IP yang mendukung komunikasi unreliable dan connectionless antara host-host dalam suatu jaringan.

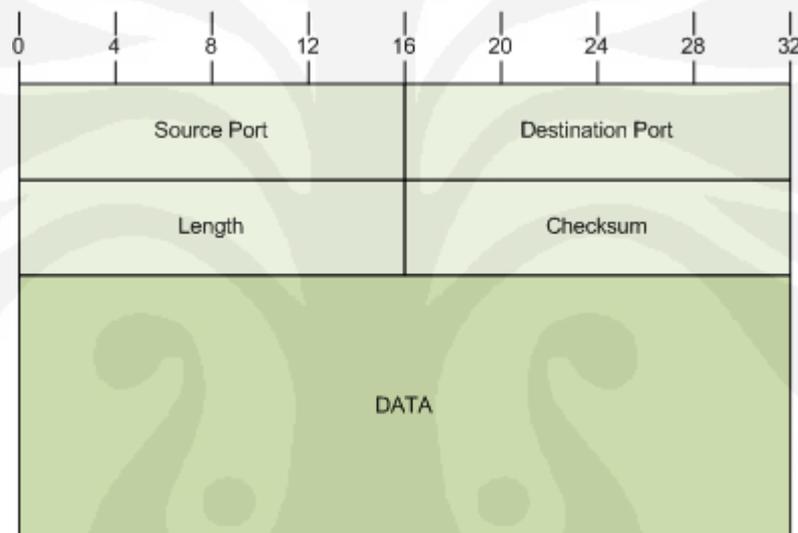
Pemilihan protocol UDP sebagai transfer data pada monitoring ini lebih kearah kemudahan penerapannya dalam bahasa pemrograman Python. Protocol UDP didefinisikan pada RFC 768. Beberapa karakteristik UDP sebagai berikut :

- a. Connectionless (tanpa koneksi), paket-paket UDP dikirimkan tanpa harus melakukan sinkronisasi terlebih dahulu dengan server.
- b. Unreliable (tidak handal), datagram yang dikirimkan tidak selalu urut.
- c. UDP menyediakan mekanisme pengiriman pesan ke sebuah protocol lapisan aplikasi atau ke proses tertentu.
- d. Adanya perhitungan checksum 16 bit pada pesan.

- e. UDP tidak menyediakan layanan buffering, segmentasi data, maupun flow-control.

Dalam aplikasinya penggunaan UDP banyak diterapkan pada aplikasi yang :

- a. Ringan (lightweight)
- b. Protocol lapisan aplikasi yang menyediakan layanan reliable (handal)
- c. Protocol yang tidak membutuhkan layanan kehandalan tinggi
- d. Transmisi broadcast



Gambar 2.1 UDP Message Format

Sumber data olahan

2.3. HARDWARE

2.3.1. Microcontroller AVR

Mikrokontroler merupakan sebuah “one chip solution” yang mana pada dasarnya suatu rangkaian terintegrasi (Integrated Circuit-IC) yang telah mengandung secara lengkap berbagai komponen pembentuk sebuah computer[14]. Berdasarkan fungsinya, mikrokontroler secara umum digunakan untuk menjalankan program yang bersifat permanen pada sebuah aplikasi yang spesifik (misal aplikasi yang berkaitan dengan pengontrolan dan monitoring).

Sedangkan program aplikasi yang dijalankan pada sistem microprosesor biasanya bersifat sementara dan berorientasi pada pengolahan data.

Berdasarkan arsitekturnya, AVR merupakan mikrokontroler RISC (Reduce Instruction Set Computer) dengan lebar bus data 8 bit. Untuk penyimpanan data, microcontroller AVR menyediakan dua jenis memori yang berbeda: EEPROM (Electrically Erasable Programmable Read Only Memory) dan SRAM (Static Random Access memory). Kapasitas simpan data kedua memori ini bervariasi tergantung pada jenis AVR-nya.

Microcontroller keluarga AVR ini muncul di pasaran dengan tiga seri utama: tinyAVR, ClasicAVR (AVR), megaAVR. AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer). Hampir semua instruksi dieksekusi dalam satu siklus clock.

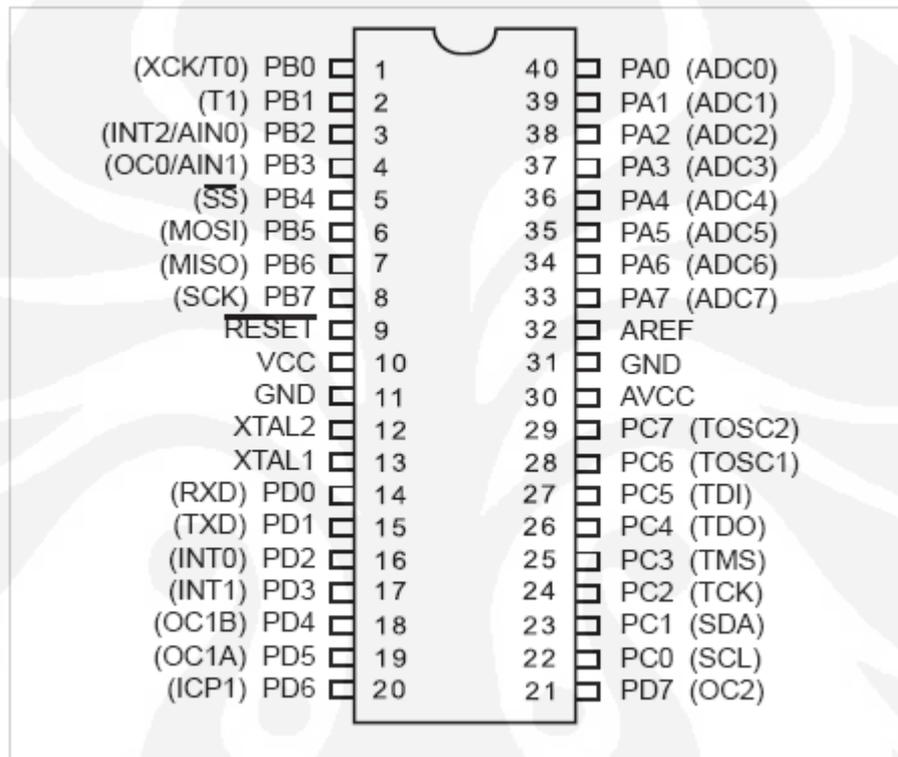
AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial UART, programmable Watchdog Timer, dan mode power saving, ADC dan PWM internal. AVR juga mempunyai In-System Programmable Flash on-chip yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI ATmega32. ATmega32 mempunyai throughput mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

ATmega32 ini merupakan salah satu tipe mikrokontroler dari seri megaAVR. Beberapa Feature-feature dari ATmega32 :

- High Performance, Low Power AVR(r) 8-bit Microcontroller
- Advance RISC architecture
- High Endurance Non-volatile memory Segment
- Dua timer 8 bit
- Satu timer 16 bit
- Realtime Counter
- Empat channel PWM
- 8 channel, 10 bit ADC
- Byte-oriented Two-wire Serial Interface

- 32 I/O Lines
- Beroperasi pada 4,5 - 5,5V

ATMega32 dikemas dalam 40 pin DIP. Koonfigurasi pin-pin ATMega32 dapat dilihat pada gambar berikut[15] :



Gambar 2.2. Konfigurasi Pin ATMega32

Sumber Datasheet ATMega32

Guna memaksimalkan performa, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data). Diskripsi dari pin-pin ATMega32 adalah sebagai berikut[14]:

- VCC : Supply tegangan digital.
- GND : Ground
- Port A : Port A sebagai input analog ke A/D konverter. Port A juga sebagai 8-bit bi-directional port I/O, jika A/D konverter tidak digunakan. Pin-pin port dapat menyediakan resistor-resistor internal pull-up. Ketika port A digunakan sebagai input dan pull eksternal

yang rendah akan menjadi sumber arus jika resistor-resistor pull-up diaktifkan. Pin-pin port A adalah tri-state ketika kondisi reset menjadi aktif sekalipun clock tidak aktif.

- d. Port B : Port B adalah port I/O 8-bit bi-directional dengan resistor-resistor internal pull-up. Buffer output port B mempunyai karakteristik drive yang simetris dengan kemampuan keduanya sink dan source yang tinggi. Sebagai input, port B yang mempunyai pull eksternal yang rendah akan menjadi sumber arus jika resistor-resistor pull-up diaktifkan. Pin-pin port B adalah tri-state ketika kondisi reset menjadi aktif sekalipun clock tidak aktif.
- e. Port C : Port C adalah port I/O 8-bit bi-directional dengan resistor-resistor internal pull-up. Buffer output port C mempunyai karakteristik drive yang simetris dengan kemampuan keduanya sink dan source yang tinggi. Sebagai input, port C yang mempunyai pull eksternal yang rendah akan menjadi sumber arus jika resistor-resistor pull-up diaktifkan. Pin-pin port C adalah tri-state ketika kondisi reset menjadi aktif sekalipun clock tidak aktif. Jika antarmuka JTAG enable, resistor-resistor pull-up pada pin-pin PC5(TDI), PC3(TMS), PC2(TCK) akan diaktifkan sekalipun terjadi reset.
- f. Port D : Port D adalah port I/O 8-bit bi-directional dengan resistor-resistor internal pull-up. Buffer output port D mempunyai karakteristik drive yang simetris dengan kemampuan keduanya sink dan source yang tinggi. Sebagai input, port D yang mempunyai pull eksternal yang rendah akan menjadi sumber arus jika resistor-resistor pull-up diaktifkan. Pin-pin port D adalah tri-state ketika kondisi reset menjadi aktif sekalipun clock tidak aktif.
- g. Reset : Sebuah low level pulsa yang lebih lama daripada lebar pulsa minimum pada pin ini akan menghasilkan reset meskipun clock tidak berjalan.
- h. XTAL1 : Input inverting penguat Oscilator dan input internal clock operasi rangkaian.
- i. XTAL2 : Output dari inverting penguat Oscilator.

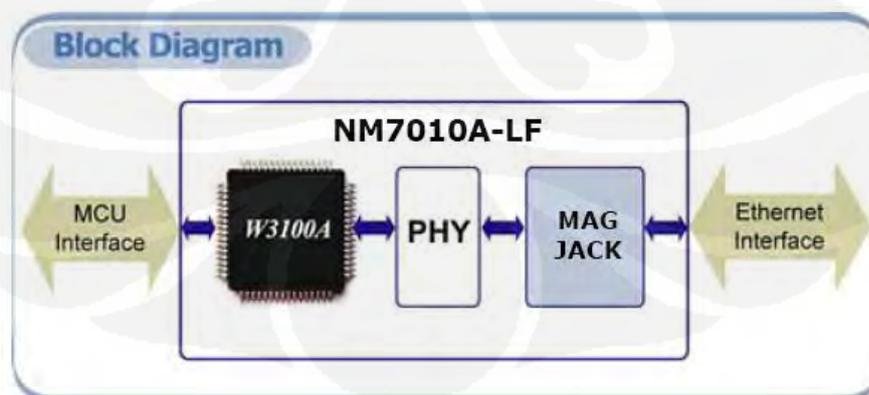
- j. AVCC : Pin supply tegangan untuk Port A dan A/D converter .
Sebaiknya eksternalnya dihubungkan ke VCC meskipun ADC tidak digunakan. Jika ADC digunakan seharusnya dihubungkan ke VCC melalui low pas filter.
- k. AREF : Pin referensi analog untuk A/D konverter.

2.3.2. Network Modul NM7010

NM7010 merupakan suatu modul Ethernet berbasis chip W3100A (TCP/IP hardwired chip) dengan beberapa tambahan berupa ethernet PHY, MAGJACK (RJ45 dan X'FMR)[16]. Beberapa fitur yang dimiliki oleh Network Module NM7010A sebagai berikut :

- Support 10/100 mbps TX
- Support half/full duplex operation
- Support Auto-Negotiation
- IEEE 802.3/802.3u complaints
- Beroperasi pada 3.3V dengan toleransi I/O 5V
- LED indicator
- I2C interface
- Socket IP

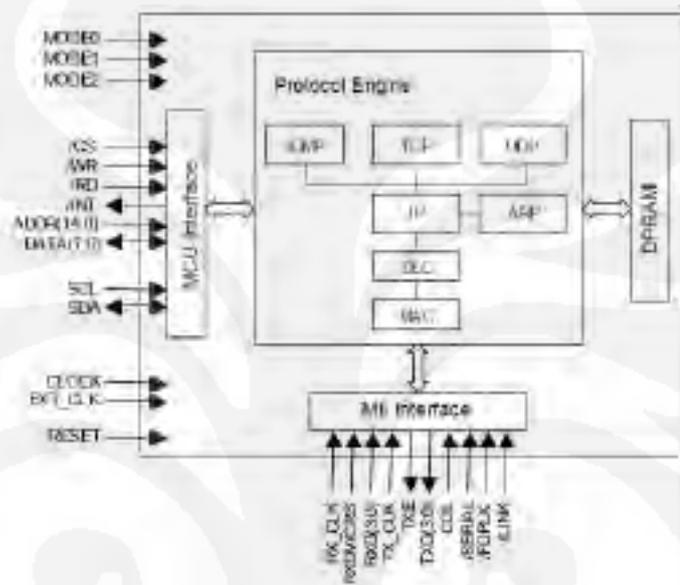
Block diagram dari Network Module NM7010A dapat dilihat seperti gambar dibawah :



Gambar 2.3. Block Diagram NM7010A

Sumber : datasheet NM7010A- IF

I2Chip W3100A merupakan hardware protocol Stack LSI yang menyediakan kemudahan koneksi jaringan dengan antarmuka I2C. Adanya antarmuka I2C, I2Chip W3100A dapat langsung dihubungkan dengan peralatan digital seperti microcontroller. Berikut Block Diagram I2Chip W3100A.



Gambar 2.4 Blok Diagram I2Chip W3100A

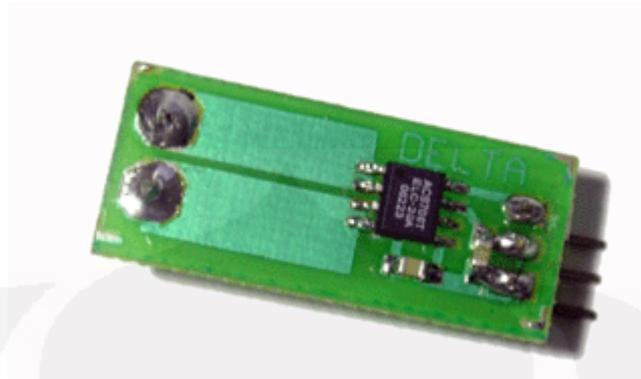
Sumber : datasheet W3100A

2.3.3. Sensor Arus DCS-01

Sensor arus yang digunakan adalah delta current sensing (DCS-01) dengan spesifikasi sebagai berikut[17] :

- 4,5 - 5,5V single supply operation
- 100mV/A output sensitivity and 20A dynamic range
- Output voltage proportional to ac and dc currents
- PCB Fiber & industrial standard

DCS-01 merupakan modul rangkaian pengkondisi yang menggunakan teknik hall effect untuk mendeteksi adanya aliran arus hingga 20A melalui modul tersebut[17]. Keluaran dari modul ini dapat dihubungkan pada ADC sehingga sistem mikrokontroler dapat menghitung nilai besaran arus yang lewat.



Gambar 2.5 Modul DCS-01

Sumber : datasheet DCS-01

Dalam pengukuran arus biasanya membutuhkan sebuah resistor shunt dimana teknologi hall effect yang diterapkan oleh Allegro menggantikan fungsi resistor shunt dan current transformer menjadi sebuah sensor dengan ukuran yang relatif jauh lebih kecil. Aliran arus listrik yang mengakibatkan medan magnet dan menginduksi bagian dynamic offset cancellation dari ACS706ELC-20A. Bagian ini akan dikuatkan oleh bagian amplifier dan melalui proses filter sebelum dikeluarkan melalui kaki 6 dan 7.

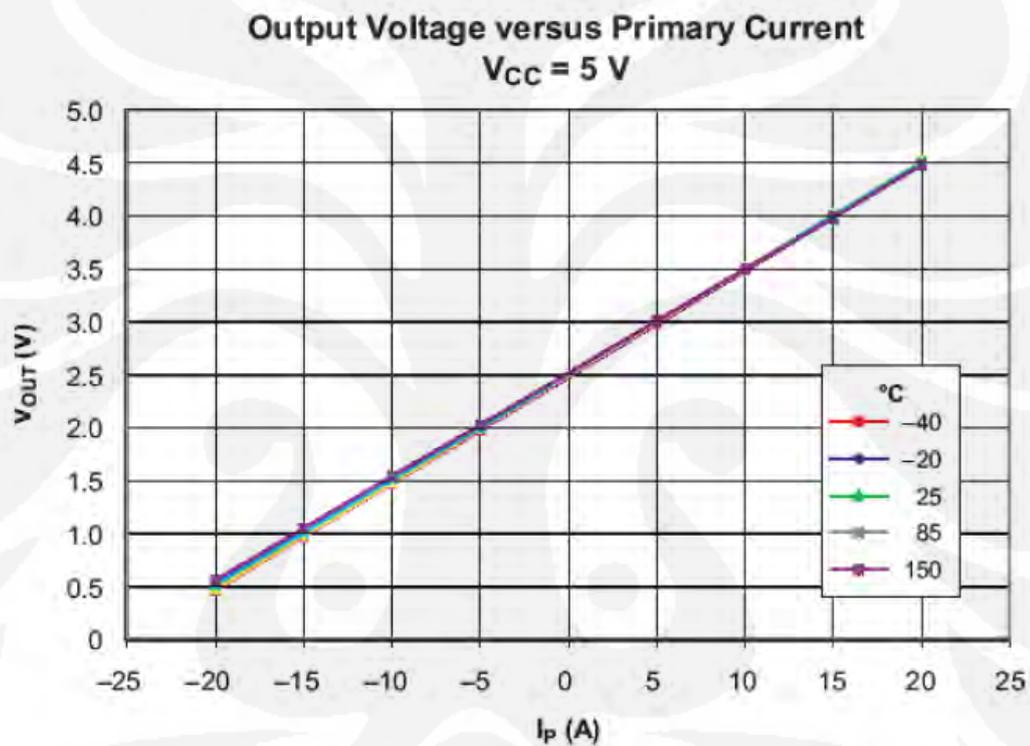
Modul DCS-01 dapat berjalan menggunakan power supply tunggal 4.5 sampai 5 volt dan dapat mengukur arus sampai dengan 20 Ampere. Untuk aplikasi pengukuran beban yang akan ditampilkan ke PC, digunakan Port RS232/USB sehingga dapat terhubung langsung ke PC, namun untuk menghubungkan modul ini dengan sistem mikrokontroler dapat digunakan Port RS232 yang merupakan level TTL. Untuk aplikasi pengukuran beban pada sistem mikrokontroler keluarga AVR, tidak dibutuhkan lagi ADC karena sistem mikrokontroler ini telah memiliki internal ADC.



Gambar 2.6 Penggunaan DCS-01 untuk pengukuran beban

Sumber datasheet DCS-01

Untuk aplikasi penggunaan beban yang akan ditampilkan ke PC, digunakan modul USB / RS-232 sehingga dapat terhubung langsung ke PC. Modul sensor DCS-01 akan mengeluarkan tegangan 0,5 volt pada saat arus yang mengalir adalah sebesar – 20 Ampere, tegangan 4,5 volt ($V_{cc} = 5$ volt) jika arus yang mengalir adalah sebesar 20 Ampere dan mengeluarkan tegangan 2,5 volt ($V_{cc}/2$) jika tidak ada arus yang mengalir.



Gambar 2.7 Grafik keluaran DCS-01 terhadap arus yang disensor

Sumber datasheet DCS-01

Modul DCS-01 dapat digunakan untuk mengukur arus searah maupun arus bolak-balik.

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Perkembangan teknologi informasi yang cepat mempengaruhi berbagai bidang, tidak ketinggalan pula di bidang sistem monitoring. Dengan adanya internet, maka teknologi informasi semakin lebih terbuka dan aksesnya menjadi lebih luas.

Issue global warming akhir-akhir ini menjadi issue hangat di dunia. berbagai penelitian mengenai sumber daya energi alternatif banyak dilakukan. Salah satunya adalah Universitas Indonesia yang mengembangkan penelitian tentang energi alternative, yaitu pemanfaatan photovoltaic.

Pada skripsi sebelumnya telah diteliti tentang "Sistem Monitoring Energi Listrik pada Photovoltaic secara Realtime"[1]. Sistem monitoring energi listrik pada Photovoltaic menggunakan komunikasi serial (RS232), serta menggunakan bahasa pemrograman Delphi dan Database Microsoft Access untuk pembuatan software monitoringnya.

Pada penulisan skripsi ini akan dicoba diteliti mengenai pemanfaatan software-software open source untuk membuat sistem monitoring dengan berbasis web. Selain itu, protokol komunikasi digunakan untuk monitoring ini adalah protokol TCP/IP. Alasan yang mendasari hal tersebut antara lain adanya kemungkinan antara sistem control dan monitoring dengan lokasi instalasi sistem photovoltaic berjauhan, oleh karena itu monitoring menggunakan komunikasi serial (RS232) menjadi kurang efektif. Alasan lainnya adalah dengan pemanfaatan software free dan open source, dimungkinkan pembuatan software dengan biaya yang cukup murah (gratis) serta dapat dikustomisasi khusus untuk monitoring.

Dalam skripsi ini, penulis akan memanfaatkan distro Linux open source Ubuntu 8.04 LTS (Long Term Support) sebagai server monitoring. Software monitoring yang dirancang berbasis web, sehingga nantinya akan dapat diakses

dari Local Area Network (LAN) maupun internet dengan menggunakan web browser.

Hardware monitoring untuk memonitor output dari photovoltaic menggunakan hasil penelitian sebelumnya mengenai “Sistem Monitoring Photovoltaic secara Realtime” dengan penambahan Modul TCP/IP.

Berdasarkan hal tersebut distro Linux Server Monitoring dibuat dengan harapan dapat menjadi sebuah aplikasi open source yang dapat digunakan pada sistem monitoring photovoltaic yang lain

1.2. TUJUAN

Skripsi ini memiliki beberapa tujuan sebagai berikut :

1. Membuat software monitoring berbasis web menggunakan software-software free dan open source.
2. Mengkustomisasi distro Linux Ubuntu 8.04 LTS untuk server sistem monitoring photovoltaic.
3. Membuat hardware monitoring untuk pengujian software monitoring.
4. Mengimplementasikan distro Linux Server dan hardware monitoring pada system photovoltaic yang ada.

1.3. RUANG LINGKUP DAN PEMBATASAN MASALAH

Ruang lingkup penulisan skripsi ini dibatasi pada

1.3.1. Ruang Lingkup

Ruang lingkup permasalahan pada penulisan skripsi ini sebagai berikut :

1. Menggunakan Ubuntu Server 8.0.4 LTS sebagai distro Linux yang akan dikustomisasi sebagai distro linux server
2. Menggunakan software-software free dan open source seperti Apache Webserver dan MySQL Database server sebagai Server berbasis web.
3. Menggunakan bahasa pemrograman Python dan PHP untuk membuat software monitoring.
4. Menggunakan NM7010A-LF network module sebagai pengirim data dengan menggunakan protocol UDP

1.3.2. Pembatasan Masalah

Permasalahan pada penulisan skripsi ini dibatasi pada :

1. Pembuatan Software Monitoring berbasis web yang akan diinstalasi pada Server monitoring
2. Pembuatan Distro Linux untuk server monitoring berbasis Ubuntu Server 8.0.4 LTS
3. Pembuatan Testboard/hardware untuk memonitor output tegangan dan arus dari sensor photovoltaic serta
4. Koneksi Testboard dengan NM7010A-LF network module sebagai pengirim data melalui protocol UDP .

1.4. METODOLOGI

Perancangan Distro Linux untuk aplikasi monitoring menggunakan metodologi :

1. Studi Literature
 - a. Linux dan pembuatan distro Linux.
 - b. Pemrograman socket untuk aplikasi server.
 - c. Pemrograman web untuk frontend.
 - d. Aplikasi sistem monitoring menggunakan microcontroller 8 bit ATmega 32.
2. Perancangan serta pembuatan software sistem monitoring.
3. Perancangan serta pembuatan hardware yang mendukung software sistem monitoring.
4. Implementasi dan pengujian pada sistem photovoltaic

1.5. SISTEMATIKA PENULISAN

Penulisan skripsi ini tersusun dari lima bab yang selanjutnya akan dijelaskan dengan sub bab. Secara keseluruhan skripsi ini disusun sebagai berikut :

1. Pendahuluan

Bab pendahuluan merupakan bab yang berisi latar belakang, tujuan, perumusan masalah, metodologi dan sistematika penulisan.

2. Landasan Teori

Bab ini menerangkan teori-teori dasar yang digunakan dalam penulisan skripsi seperti : linux dan open source, bahasa pemrograman yang digunakan dalam pembuatan software, hardware monitoring yang menggunakan microcontroller ATmega 32, sensor tegangan dan sensor arus, serta firmware monitoring.

3. Perancangan Aplikasi Monitoring dan Hardware Monitoring

Bab ini menjelaskan perancangan software monitoring dan hardware monitoring.

4. Implementasi dan Pengujian

Bab ini menggambarkan mengenai langkah-langkah kustomisasi distro Linux Ubuntu 8.04 LTS sebagai server monitoring, implementasi distro linux server monitoring dan hardware monitoring pada system photovoltaic.

5. Penutup

Bab ini berisi kesimpulan yang didapat dari skripsi serta saran yang membangun untuk penelitian selanjutnya

DAFTAR REFERENSI

- 1 Silberschat. (2003). “Operating System Concepts – Fourth Edition”, John Willey & Son
- 2 Sofyan, Ahmad. (2006). “Membuat distro Sendiri 0.6.3”. Dian Rakyat.
- 3 Wikipedia. (2010). Ubuntu, <http://id.wikipedia.org/Ubuntu> (diakses 21 Mei 2010).
- 4 Resmaster Ubuntu untuk School Onffline menggunakan UCK.(2010). http://opensource.telkomspeedy.com/wiki/index.php/Remaster_Ubuntu_untuk_SchoolOnffLine_Menggunakan_UCK onlie diakses tanggal 3 Mei 2010.
- 5 Wikipedia. (2010). Python, http://id.wikipedia.org/Apache_HTTP_server (diakses 21 Mei 2010).
- 6 Apache docs. (2010). <http://httpd.apache.org/docs/2.2/> (diakses 20 April 2010)
- 7 Wikipedia. (2010). Python, <http://id.wikipedia.org/MySQL> (diakses 21 Mei 2010).
- 8 Solichin, Ahmad .(2010). “MySQL 5 dari Pemula hingga mahir”. ACHMATIM.NET
- 9 Azis, Farid. (2005). “Object Oriented Programming dengan PHP5”, Elek Media Komputindo
- 10 Wikipedia. (2010). UDP, <http://id.wikipedia.org/Python> (diakses 21 Mei 2010).
- 11 Daemon in Python (2010) <http://code.activestate.com/recipes/278731-creating-a-daemon-the-python-way/> online (diakses 15 Mei 2010)
- 12 Stevens, Comer. (1997). “Client Server Programming and Application”, Prentice Hall.
- 13 Wardhana, Lingga. (2006). “Microcontroller AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi”, Penerbit ANDI.

- 14 ATMEL. (2010) datasheet ATmega32L, (online), http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf (diakses tanggal 2 April 2010)
- 15 Datasheet Network Module NM7010A-LF (2010) <http://www.kosmodrom.com.ua/data/wiznet/NM7010ALF%20Datash eet%20V2.6.pdf> online (diakses tanggal 26 Mei 2010)
- 16 Manual DCS-51 (2010) <http://deltakits-sby.com/product/dcs01/manual.pdf> online (diakses tanggal 15 Mei 2010)