

# KONFIGURASI OTOMATISASI SHUTDOWN DAN MONITORING POWER PADA SERVER MENGGUNAKAN SMART CARD UPS-APC

Siswo Nugroho<sup>\*)</sup>, Sudjadi, and Adian Fatchur Rochim

Jurusan Teknik Elektro, Universitas Diponegoro Semarang  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*)</sup>E-mail : kalongklik.scooterz@gmail.com

## Abstrak

Seiring dengan berkembangnya teknologi informasi, maka kebutuhan informasi manusia pun meningkat. Teknologi Informasi saat ini sedang berkembang pesat, dan pada dasarnya manusia membutuhkan segala sesuatu yang semakin mudah dan praktis tanpa harus banyak membuang tenaga serta biaya namun dapat memberikan hasil yang maksimal secara cepat dan akurat. Salah satu layanan yang dapat diimplementasikan dengan teknologi IT adalah Konfigurasi Otomatisasi Shutdown Dan Monitoring Power Pada Server Menggunakan Smart Card UPS-APC. Pada Tugas Akhir ini menggunakan perangkat keras berupa UPS-APC sebagai backup power untuk server dan sistem operasi ubuntu server untuk konfigurasi monitoring dan otomatisasi pematikan server. UPS-APC mempunyai banyak fitur yang dapat digunakan untuk pengembangan Tugas Akhir ini, khususnya pemanfaatan fitur Smart Signaling pasokan daya dan tegangan yang di suplai untuk power server. Hasil yang didapatkan dari proyek tugas akhir ini adalah monitoring dan otomatisasi ketika power untuk server mengalami kegagalan sehingga server memiliki tindakan untuk mematikan dirinya sendiri dan memudahkan untuk menentukan power yang sehat untuk server.

*Kata Kunci : monitoring, otomatisasi, UPS-APC, Smart Card.*

## Abstract

As Along as with the development of information technology, it needs people to increased. The Information technology is now being developed rapidly, and basically the people need everything practical that easier without having to waste a lot of effort and cost but can deliver maximum result are quickly and accurate. One of the services that can be implemented with IT technology are Shutdown And Automation Configuration Monitoring Power on the Server Using Smart cards UPS-APC. On this final project uses hardware in form of ups-apc as backups power for server, an operating system of ubuntu server to the configuration of monitoring, and shutdown server automatitaton. The ups-apc have many features that can be used for development the final project, specially utilization of smart signaling features the supply of power and a voltage in supply for the power server. The results obtained from this final project are monitoring and automatitaton when the power to the server so that the server has experienced a failure action to turn off itself and makes it easier to determine the power of a good for servers.

*Key words: monitoring, automatitaton , the UPS-APC, the Smart Card.*

## 1. Pendahuluan

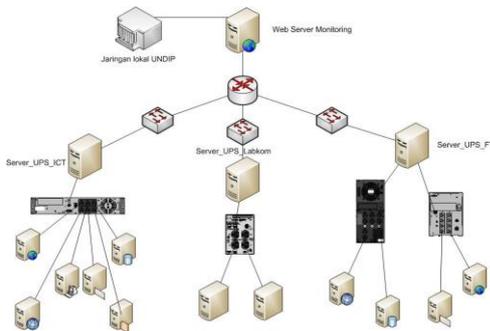
UNDIP merupakan salah satu universitas yang membutuhkan sebuah sumber tenaga yang tak terputus atau UPS (*Uninterruptible Power Supply*) untuk memenuhi keperluan beberapa komputer *server* utama yang mengkomputasi beberapa kebutuhan penting seperti Sistem Informasi Akademik, Pembayaran Perkuliahan Mahasiswa, dan lain-lain. Banyaknya pengguna internet yang melakukan pengaksesan konten baik itu gambar maupun video, memerlukan komputer *proxy servers* untuk mengatur dan menyaring keluar masuknya konten yang di akses.

Pengendalian terhadap UPS dapat dilakukan dengan menggunakan *smart-card* yang telah di sediakan. *Smart Card* digunakan untuk mengetahui *signal* dari *UPS-APC* yang nantinya dari data tersebut akan dikenali keterangan berupa *power in* dan *power out* yang dibutuhkan komputer server untuk beroperasi secara normal. *Signal* digunakan untuk mengirim perintah *shutdown* ketika terjadi kegagalan kelistrikan dan pada saat kelistrikan kembali normal akan menerima perintah *power on*. Perangkat keras ini juga memanfaatkan *signal* arus searah yang akan diintegrasikan dengan jaringan *LAN* yang digunakan sebagai monitoring pada setiap komputer server yang terhubung pada UPS-APC.

## 2. Metode

Perancangan ini ditujukan untuk implementasi monitor *UPS APC* di Universitas Diponegoro oleh karena itu lokasi *server* harus berada dalam satu perangkat *UPS* yang terhubung dengan jaringan internet, baik secara privat (jaringan yang alamat IP nya tidak perlu diketahui oleh jaringan internasional) ataupun publik.

Penggunaan alamat IP juga akan disesuaikan dengan alokasi alamat IP jaringan komputer UNDIP. Desain *server* menggunakan sebuah mesin menggunakan sistem operasi CentOS 6.4 *server* yang terkoneksi dengan jaringan privat Universitas Diponegoro. Sistem pada *server* meminta paket data penggunaan daya listrik yang disediakan oleh *Smart-UPS* tersebut.



Gambar 1. Perancangan sistem secara umum

### 2.1 Perancangan Alarm Notifikasi Email

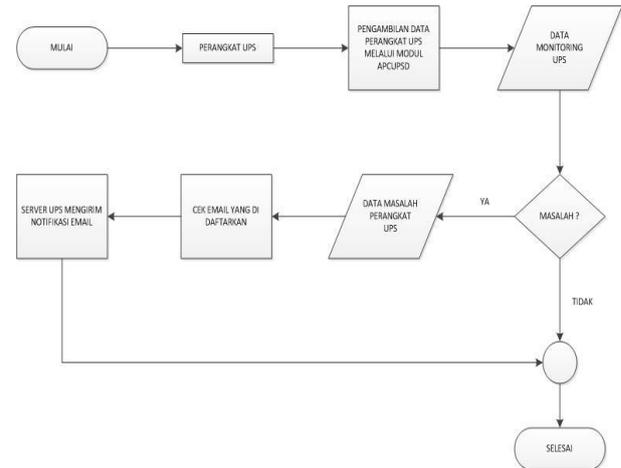
Perancangan notifikasi *email* pada tugas akhir kali ini memanfaatkan *Simple Mail Transfer Protocol (SMTP)* sebagai *mail-server* untuk mengirimkan notifikasi gangguan yang terjadi pada perangkat *UPS*. Pada perancangan tugas akhir kali ini digunakan layanan SMTP yang disediakan oleh penyedia layanan *email google*. Penggunaan layanan SMTP *google* dikarenakan perancangan tugas akhir ini tidak menanamkan sistem *mail-server* dalam *serverUPS*. *Server UPS* secara otomatis akan mengirim detail masalah melalui *email* kepada teknisi yang telah terdaftar pada *server UPS* melalui protokol SMTP yang telah diautentifikasi oleh *mail-server google*.

*Flowchart* pada gambar 3.10 dapat dijelaskan sebagai berikut:

1. Perangkat yang dimonitor mengirimkan data ke *server UPS*
2. Data yang di dapat diolah oleh *Server UPS*.
3. Jika terjadi masalah maka data akan dikirimkan kepada kepala teknisi melalui *email* yang telah di daftarkan.

Pengiriman notifikasi *email* dilakukan oleh *server UPS* melalui layanan SMTP dengan mengirimkan detail

masalah gangguan yang terjadi pada perangkat *UPS* yang dimonitor oleh *server*.



Gambar 2. *Flowchart* pengiriman notifikasi melalui *email*

### 2.2. Perancangan Tampilan Aplikasi Monitoring dengan *Apcupsd-cgi*

Faktor yang menjadi unggulan pada perancangan tugas akhir ini adalah perancangan tampilan aplikasi *monitoring* dengan *apcupsd-cgi* yang merupakan dukungan *monitoring*. Tampilan ini akan mempermudah teknisi pada pemantauan *webserver* yang akan menampilkan menu untuk melakukan perawatan. Aplikasi *apcupsd-cgi* akan membantu menghasilkan grafik *UPS* dalam bentuk gambar berformat *.PNG* yang juga berguna untuk mengetahui letak permasalahan yang terjadi bila terjadi gangguan. *Apcupsd-cgi* juga dapat menampilkan grafik kapasitas baterai, tegangan baterai, tegangan keluaran *UPS*, dan utilitas *UPS*.

Akses *login* pada *webserver* hanya memperbolehkan administrator dan teknisi, untuk melakukan modifikasi pada pemetaan *monitoring* perangkat *UPS*. Hal ini dilakukan karena akses yang terbatas perangkat *UPS* hanya dilakukan oleh administrator dan teknisi, dan juga dikarenakan teknisi lebih mengetahui tentang perawatan yang diterapkan.

Perancangan pemetaan perangkat dilakukan dengan menambahkan perangkat *server UPS* terlebih dahulu kemudian memasukkan data perangkat *UPS* yang sesuai dengan peta letak perangkat *UPS* yang telah dibuat.

### 2.3. Perancangan Aplikasi Monitoring *UPS*

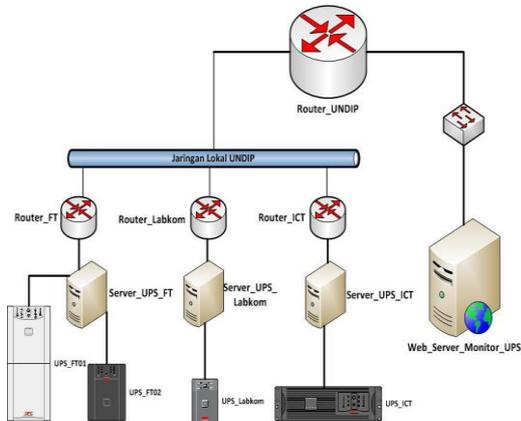
Aplikasi *Monitoring UPS* dibangun dengan menggunakan bahasa pemrograman PHP, MySQL sebagai database user, dan PHP MyAdmin yang dapat di program ulang, yang akan mengatur EEPROM *Universal Serial BUS UPS*, maka aplikasi *server monitoring UPS* harus dihubungkan dengan *server UPS* yang sudah di tanam paket *apcups daemon* dan *apcups-cgi*. *Apcups daemon* berfungsi sebagai monitor yang berguna untuk

mengambil data UPS melalui EEPROM yang di hubungkan dengan USB (*Universal Serial Bus*). *Apcupsd-cgi* berfungsi untuk menampilkan data yang sudah diolah kedalam bentuk grafik dan gambar yang di tampilkan melalui *webserver*. Selain itu juga ditanamkan program *shell\_exec* pada halaman web berbasis php yang akan digunakan untuk memerintahkan *server* lain *off* ketika UPS sudah mendekati batas akhir waktu hidup

Desain *server* UPS menempatkan *server* pada posisi terkoneksi dengan UPS dan jaringan lokal UNDIP. Hal ini didasarkan pada fungsi utama *webserver* UPS yaitu untuk memonitoring perangkat dari beberapa *server* UPS yang menyediakan sumber daya listrik *server* sistem informasi UNDIP.



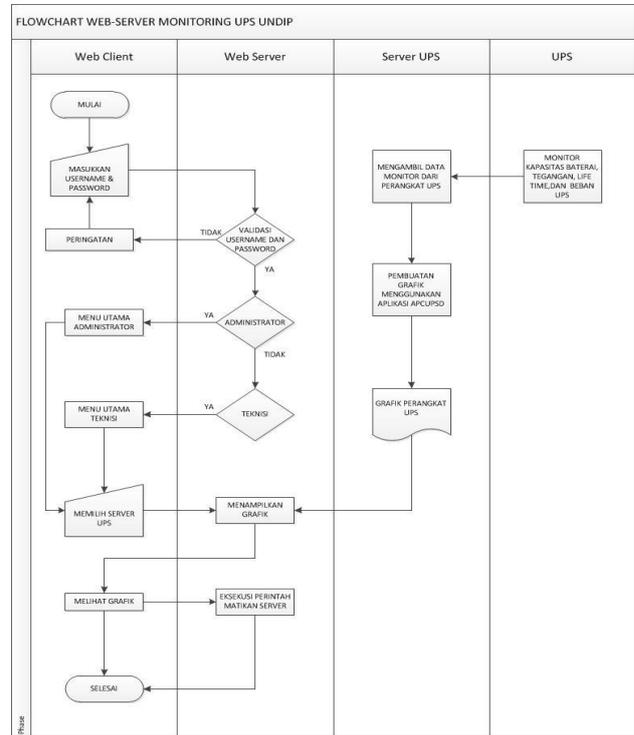
Gambar 3. Skema pembuatan peta UPS



Gambar 4. Skema aplikasi monitor UPS

Pengecekan terhadap perangkat UPS pada aplikasi akan dipantau oleh modul *plug-in apcupsd-cgi*. Hasil analisa

dari modul *apcupsd-cgi* selanjutnya akan dikirim pada *server* untuk divisualisasikan oleh aplikasi. *Output* yang dihasilkan oleh aplikasi adalah *file* gambar berformat PNG. Berikut ini adalah *flowchart* sistem aplikasi monitoring UPS UNDIP :



Gambar 5. Flowchart sistem monitoring

Flowchart pada gambar 3. dapat dijelaskan sebagai berikut:

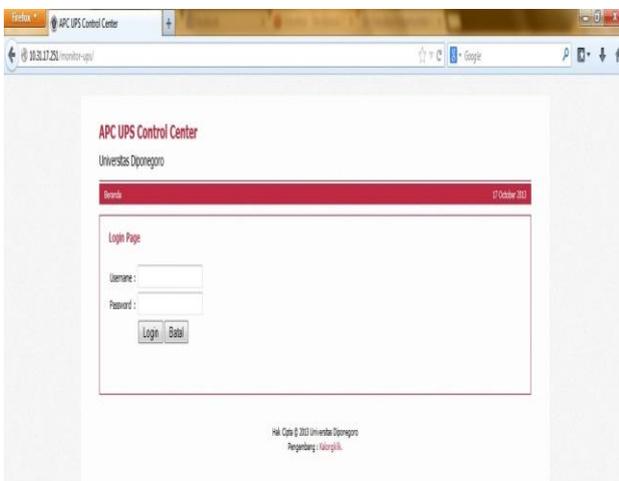
1. Setelah *user* melakukan akses ke *webserver* melalui *web client* maka akan tampil *weblogin server ups*.
2. Memasukkan *username* dan *password* pada halaman *weblogin*.
3. *Server* akan melakukan *validasi user*. Jika *user* belum terdaftar maka akan muncul peringatan.
4. Jika *user* yang melakukan *login* terdaftar pada *database* maka *server* akan mengijinkan *user* masuk. Namun disini juga akan diperiksa apakah *user* yang melakukan *login* merupakan administrator atau teknisi.
5. Jika *user* yang melakukan *login* merupakan administrator maka *server* akan mengarahkan ke halaman administrator. Jika *user* teknisi maka *server* akan menampilkan menu tampilan teknisi.
6. Grafik diperoleh dari pengambilan data yang dilakukan *webserver* kepada perangkat melalui modul *plug-in apcupsd-cgi* yang kemudian data tersebut dikelola menjadi gambar grafik dengan format PNG.
7. Setelah grafik sudah terbentuk maka grafik akan dikirimkan oleh *server UPS* kepada *web server UPS* untuk ditampilkan kepada *user* yang mengakses data tersebut.

### 3. Hasil dan Analisa

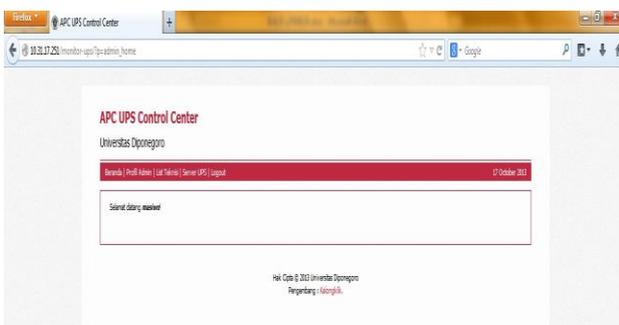
Implementasi dilakukan dengan membuat layanan-layanan seperti pada perancangan. Dimulai dengan penanaman sistem operasi pada *server* sampai dengan instalasi aplikasi-aplikasi yang diperlukan oleh *server*. Setelah semua paket telah terpasang, maka hal terpenting yang harus dilakukan adalah melakukan konfigurasi agar semua proses pemantauan UPS bisa berjalan seperti yang diharapkan.

#### 3.1. Pengujian Aplikasi Web Server UPS

Pengujian menggunakan *web browser*, pengguna diarahkan pada halaman *login web server*, *web server* akan mengenali *user* yang terdaftar pada database dan mengarahkan *user* sesuai *level user*, bila *user* terdaftar sebagai administrator maka *web server* mengarahkan *user* kepada menu web administrator, dan bila *user* sebagai teknisi maka *web server* mengarahkan user pada menu teknisi.



Gambar 6. tampilan halaman login web-server UPS

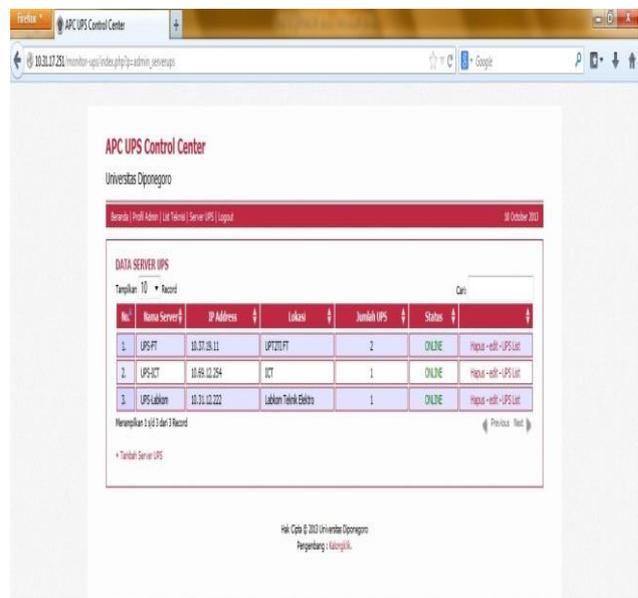


Gambar 7. tampilan login menu administrator web-server UPS



Gambar 8. Tampilan login menu teknisi web-server UPS

Kebutuhan utama dari aplikasi *monitoring* terdapat pada menu server UPS. Menu *server ups* digunakan untuk memetakan perangkat UPS yang terhubung dengan *server ups*, setiap *server UPS* dapat terhubung dengan beberapa perangkat UPS.

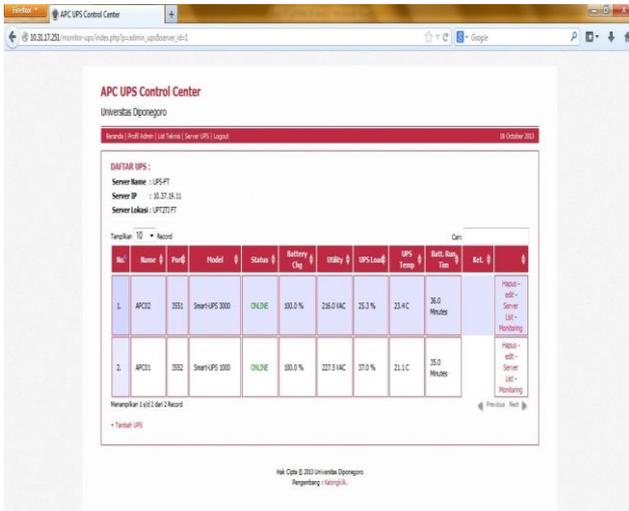


Gambar 9. tampilan menu daftar server UPS

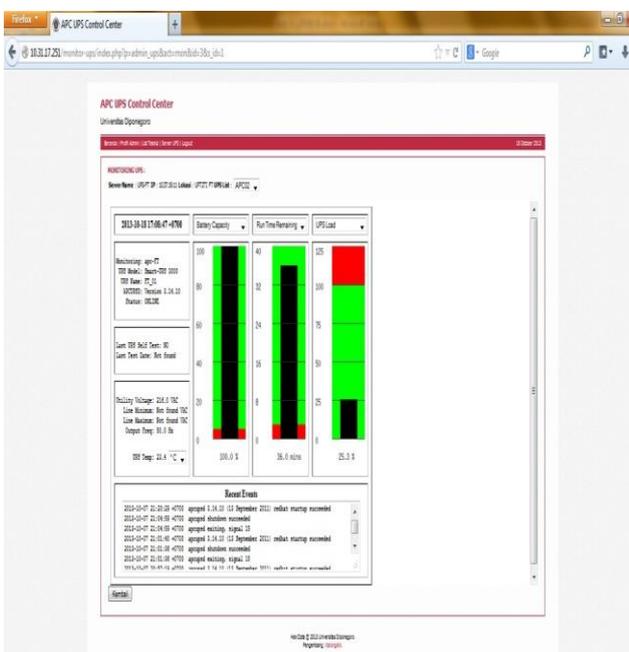
Tabel-tabel menu *server UPS* dilengkapi dengan daftar alamat *host* dari *server UPS* mempermudah dalam manajemen dalam jaringan lokal Universitas Diponegoro. Pada tabel juga dilengkapi daftar lokasi *server UPS* dan jumlah UPS yang terhubung pada setiap *server UPS*. Status *server UPS* juga disediakan sebagai monitor terhadap *server UPS*.

UPS *list* pada menu *server ups* membantu pengelola mengetahui keadaan perangkat yang terhubung dengan *server*, hal ini merupakan kebutuhan utama dari sebuah

aplikasi *monitoring*. *Monitoring* terhadap UPS dibutuhkan untuk mengetahui keadaan UPS yang menyediakan daya listrik.



Gambar 10. tampilan menu daftar UPS

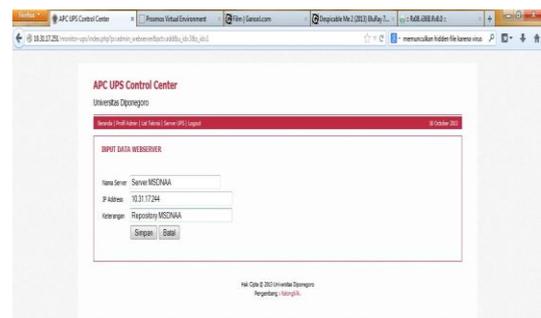


Gambar 11. tampilan grafik UPS pada aplikasi *monitoring* monitor

Menu *list* UPS menyediakan alamat *host server client* yang terhubung dengan jaringan listrik UPS, sehingga ketika terjadi gangguan *user* dapat memerintahkan *server* untuk *shutdown*, pada saat kapasitas penyimpanan daya UPS sudah mendekati masa kritis.



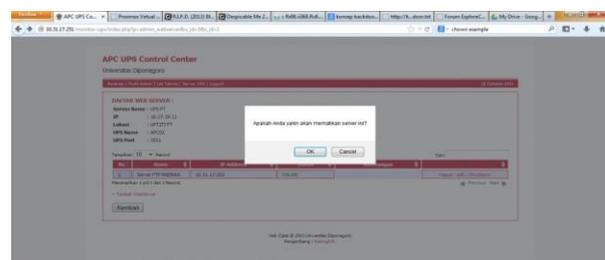
Gambar 12. tampilan *server* yang terhubung dengan UPS pada aplikasi *monitoring* monitor



Gambar 13. tampilan menu tambah *server* pada aplikasi *monitoring* monitor

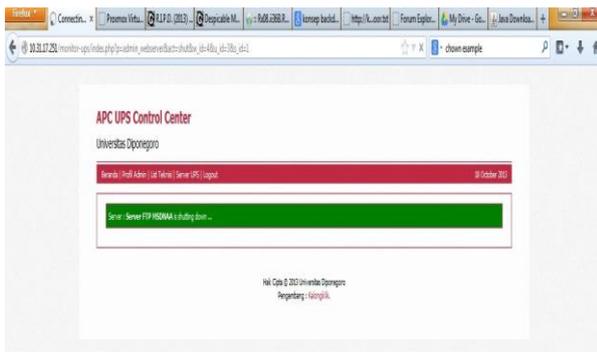
### 3.2. Pengujian Remote Shutdown pada Server Client

Pengujian eksekusi perintah *shutdown* dilakukan dengan menggunakan aplikasi *monitoring*, pada setiap user diberikan hak akses yang sama untuk melakukan eksekusi.

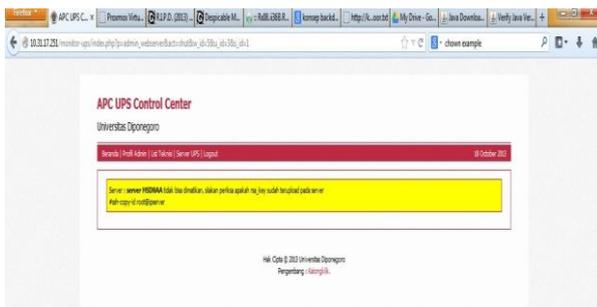


Gambar 14. tampilan notifikasi *shutdown* pada aplikasi *monitoring* monitor

Aplikasi *monitoring* memberikan *alert* apakah user ingin mematikan server, pengujian berhasil ketika *web server* memberikan notifikasi *server client* telah berhasil di *shutdown* dan bila server tidak dapat dimatikan maka server akan memberikan notifikasi bahwa *certified rsa* belum di otoritaskan kepada *server* tersebut.



Gambar 15. tampilan *shutdown server client* pada aplikasi *monitoring monitor*



Gambar 16. tampilan notifikasi peringatan *shutdown server client* pada aplikasi *monitoring monitor*

#### 4. Kesimpulan

Aplikasi *monitoring* UPS dapat membantu mengatasi masalah tentang kebutuhan akan aplikasi pemantau UPS yang handal untuk digunakan memantau proses kerja, pengambilan data kondisi UPS, dan pemetaan UPS pada jaringan koneksi lokal Universitas Diponegoro untuk digunakan sebagai pelaporan oleh teknisi terkait.

*Apcups Daemon* merupakan *plugin* utama yang digunakan dalam proses pembacaan data UPS yang dilakukan oleh aplikasi *monitoring* terhadap perangkat UPS yang dimonitoring.

#### Referensi

- [1] Kropelin, Adam. *APCUPS Daemon* "http://www.apcupsd.org/manual/manual.html".2009
- [2] Ramadhan, Arief., Hendra, Saputra., *PHP5 dan MySQL*, Jakarta Pusat, Elex Media Komputindo, 2005.
- [3] Nugroho, B., *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*, Cetakan Pertama, 2004.
- [4] Nakhjiri., *AAA based Keying for Wireless Handovers: Problem Statement*, Network Working Group, 2006.
- [5] Chandorkar, Mukul C., "*Distributed Uninterruptible Power Supply Systems*", University of Wisconsin--Madison, 1995.
- [6] Asghar, M. S. Jamil., *Power Electronics*, Prentice-Hall of India Private Limited, 2004.

- [7] Kiehne, H.A., *Battery Technology Handbook*, Expert Verlag, 2003.
- [8] Bergveld, H.J., Kruijt, W.S.,Notten, Peter H. L., *Battery Management Systems: Design by Modelling*, Kluwer Academic Publishers, 2002.
- [9] Crompton, Thomas Roy., *Battery Reference Book,Third Edition*, Reed Educational And Professional Publishing Ltd, 2000.
- [10] Sklar, D., *Learning PHP 5*, O'Reilly. June 2004
- [11] Lerdorf. R., P. MacIntyre., and K. Tatroe., *Programming PHP, 2nd Editon*, O'Reilly. April 2006.
- [12] Hayder. H., *Object-oriented Programming with PHP5*, Desember 2007.
- [13] Solichin, Achmad., *MySQL 5 : Dari Pemula Hingga Mahir*. jakarta, 2010
- [14] Welling, Luke & Thomson, Laura., *PHP and MySQL Web Development*, Sams Publishing, 2001