

# **Laporan Penelitian**

## **PEMANFAATAN TEKNOLOGI UPS UNTUK PENGELOLAAN DAYA SERVER**



oleh  
**WAGITO**

**Dilaksanakan Atas Bantuan Dana  
Pusat Penelitian dan Pengembangan  
Semester Ganjil Tahun Akademik 2010/2011**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
AKAKOM YOGYAKARTA  
2011**

## Halaman Pengesahan

1. a. Judul Penelitian : Pemanfaatan Teknologi UPS  
Untuk Pengelolaan Daya Server  
b. Macam Penelitian : Aplikasi Jaringan Komputer
2. Identitas Peneliti  
a. Nama : Wagito, S.T., M.T.  
b. Jenis Kelamin : Laki-laki  
c. NPP : 961080  
d. Pangkat/Golongan : Pembina / IV A  
e. Fakultas/Jurusan : Teknik Informatika
3. Lokasi Penelitian  
a. Tempat : STMIK AKAKOM Yogyakarta  
b. Alamat : Jl. Raya Janti,  
Karang Jambe, Yogyakarta
4. Waktu Penelitian : 6 bulan
5. Biaya Penelitian :

Yogyakarta, Juli 2011

Peneliti

Wagito, S.T., M.T.  
NPP. 961080

Mengetahui

Ketua STMIK AKAKOM

Ka. Puslitbang PPM STMIK AKAKOM

Sigit Anggoro, S.T., M.T.  
NPP. 971104

Dra. Syamsu Windarti, M.T., Apt  
NIP. 19660710 199303 2 001

## **Kata Pengantar**

Puji syukur saya panjatkan ke hadirat Allah S.W.T. karena hanya dengan rahmat dan hidayah-Nya saya dapat menyelesaikan penelitian ini. Penelitian yang berjudul Pemanfaatan Teknologi UPS Untuk Pengelolaan Daya Server ini dikembangkan dengan tujuan memanfaatkan daya listrik yang berasal dari UPS secara optimal mungkin untuk mengelola server pada saat sumber listrik utama mengalami gangguan. Dalam penelitian ini digunakan satu UPS yang punya kapasitas tinggi. Pada saat server terhindar dari gangguan atau pun kerusakan perangkat lunak dan perangkat keras.

Penulis menyadari bahwa hasil penelitian ini masih banyak kekurangannya, sehingga kritik dan saran yang membangun untuk lebih mengembangkan hasilnya sangat diharapkan. Semoga hasil penelitian ini bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Penulis

## **Daftar Isi**

Halaman Judul.....	
Halaman Pengesahan.....	ii
Kata Pengantar.....	iii
Daftar Isi.....	iv
Daftar Gambar.....	vii
Intisari.....	viii
BAB I. PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Tujuan.....	7
C. Manfaat Penelitian.....	8
BAB II. TINJAUAN PUSTAKA.....	9
BAB III. TEORI.....	12
A. Server.....	12
B. UPS (Uninterruptible Power Supply).....	16

C. WoL (Wake on LAN).....	20
D. Ether-Wake.....	25
E. Sshpass.....	27
F. Rups.....	31
BAB IV. METODOLOGI.....	38
A. Bahan Penelitian.....	38
B. Alat.....	38
C. Jalan Penelitian.....	39
BAB V. ANALISIS DAN PEMBAHASAN.....	42
BAB VI. KESIMPULAN DAN SARAN.....	54
A. Kesimpulan.....	54
B. Saran-Saran.....	55
Daftar Pustaka.....	56
LAMPIRAN.....	L-1
Isi File rupslog.....	L-1
Cara Instal Rups.....	L-2
Help Sshpass.....	L-9
Help Ether-wake.....	L-10

Curriculum Vitae.....	L-11
Personalia Penelitian.....	L-12
Biaya Penelitian.....	L-13
Jadwal Penelitian.....	L-14
Surat Keputusan.....	L-15

## **Daftar Gambar**

Gambar 1. Konfigurasi Perangkat Keras.....	39
--	----

## Intisari

Server merupakan pusat layanan dalam sistem jaringan komputer. Karena bertugas melaksanakan layanan-layanan sistem jaringan maka server harus menyala terus-menerus. Sumber daya listrik untuk server harus selalu ada dan tidak boleh terputus. Putusnya sumber daya listrik secara mendadak selain mengganggu layanan jaringan komputer juga bisa menyebabkan sistem server mengalami kerusakan. Server harus punya kesempatan untuk menjalani *shutdown* secara normal. Penyelesaian untuk kasus ini adalah penggunaan teknologi UPS. Kalau UPS dipasang pada tiap server, investasi yang diperlukan sangat besar. Diperlukan suatu cara memanfaatkan satu UPS untuk seluruh server. Diperlukan komunikasi antara UPS dengan server serta komunikasi antar server. Komunikasi ini digunakan untuk mengelola daya UPS.

Pada penelitian ini dikembangkan suatu cara untuk memanfaatkan satu UPS untuk memberi catu daya pada beberapa server. Server yang tidak perlu hidup terus akan dimatikan pada saat listrik padam, sedangkan server yang harus tetap hidup akan tetap dipertahankan selama mungkin untuk melayani sistem jaringan. Perangkat keras yang digunakan dalam penelitian adalah UPS, server dan *workstation* serta beberapa utilitas sistem jaringan komputer. Perangkat lunak yang digunakan dalam penelitian adalah Sistem Operasi Linux, Ether-Wake, Rups dan Sshpass.

Kesimpulan pada penelitian ini adalah bahwa secara umum sistem dapat bekerja secara baik, dapat menangkap isyarat yang dikirim oleh UPS. Perangkat lunak Sshpass dapat dipakai sebagai sarana untuk mematikan server dari jarak jauh. Perangkat lunak Ether-Wake dapat bekerja secara baik untuk menghidupkan server dari jarak jauh dari keadaan mati.

**Kata kunci:** daya, server, UPS, Ether-Wake, Sshpass

# **BAB I. PENDAHULUAN**

## **A. Latar Belakang**

Jaringan komputer merupakan kumpulan banyak komputer serta peralatan lain yang terhubung menggunakan media komunikasi tertentu. Komputer-komputer dalam sistem komputer dapat saling tukar-menukar data serta berbagi sumber daya yang ada pada masing-masing komputer. Sistem jaringan komputer dapat menjalankan fungsi komunikasi data dan berbagi sumber daya karena adanya layanan-layanan yang disediakan dalam sistem jaringan tersebut. Layanan yang disediakan sistem jaringan komputer misalnya *e-mail*, *chatting*, *browsing* dan sebagainya. Layanan-layanan tersebut bagi para pengguna jaringan komputer sangat memudahkan serta mempercepat penyelesaian pekerjaan yang memerlukan data dari berbagai sumber yang letaknya terpisah. Data yang terletak pada mesin-mesin yang berbeda dan terpisah dapat secara mudah serta cepat diambil melalui sistem jaringan komputer.

Server merupakan pusat layanan dalam sistem jaringan komputer. Satu server dapat melayani banyak layanan sekaligus atau bisa juga untuk tiap layanan ditangani oleh satu server. Beberapa layanan yang ringan dapat ditangani oleh satu server yang sama. Untuk layanan yang berat diperlukan satu server khusus.

Penentuan server apa melaksanakan tugas layanan apa didasarkan pada berat atau ringan layanan tersebut. Intinya dalam hal ini adalah bahwa server bertugas melaksanakan layanan-layanan jaringan komputer supaya fungsi jaringan dapat berjalan secara optimal.

Karena bertugas melaksanakan layanan-layanan sistem jaringan maka server harus menyala terus-menerus (tidak boleh mati). Apabila server mati maka tentu saja layanan akan terhenti dan dapat menyebabkan gangguan bahkan kerugian pada para pengguna. Sebagai contoh, misalnya server layanan *e-mail* yang sudah punya tingkat dunia seperti mail.yahoo.com mengalami mati, tentu saja para pengguna di seluruh dunia akan terganggu. Supaya dapat menyala terus-menerus maka server harus punya spesifikasi khusus yang dirancang supaya dapat menyala terus-menerus misalnya berkaitan dengan spesifikasi *mainboard*, prosesor, *hard disk*, RAM, kartu jaringan, *chasing* serta tidak kalah penting adalah peralatan yang bertugas memberi sumber daya listrik.

Apabila spesifikasi server sudah sesuai dan dirancang untuk dapat menyala terus menerus, sumber daya listrik juga harus disediakan secara terus-menerus. Artinya sumber daya listrik untuk server harus selalu ada dan tidak boleh terputus. Putusnya sumber daya listrik secara mendadak selain mengganggu layanan jaringan komputer juga bisa menyebabkan sistem server mengalami kerusakan. Sumber daya listrik yang tiba-tiba terputus dapat menyebabkan penyimpanan data terganggu bahkan bisa menyebabkan perangkat lunak yang digunakan untuk melakukan tugas pelayanan mengalami kerusakan. Yang paling fatal adalah

putusnya sumber daya listrik secara mendadak bisa menyebabkan rusaknya perangkat keras server terutama simpanan sekunder *hard disk*. Server harus dimatikan secara normal supaya tidak mengalami gangguan atau kerusakan.

Ketika daya listrik terputus, server harus punya kesempatan atau diberi kesempatan untuk menjalani *shutdown* secara normal supaya terjaga dari kerusakan baik perangkat lunak maupun perangkat keras. Penyelesaian untuk kasus ini adalah penggunaan teknologi UPS (*uninterruptible power supply*) yang dilengkapi dengan perangkat komunikasi dengan server untuk pengiriman isyarat bahwa daya listrik terputus sedemikian, sehingga server dapat segera melakukan proses *shutdown* secara normal. Cara seperti ini memerlukan perangkat UPS yang dipasang pada tiap-tiap server. UPS yang dapat mempertahankan daya listrik beberapa saat harganya tidak terlalu mahal dan cara ini dapat dikatakan cukup layak diterapkan. Apabila yang diinginkan hanya membuat server dapat selalu *shutdown* dengan normal, penggunaan UPS untuk tiap server dapat dikatakan sudah dapat menyelesaikan masalah.

Namun bagaimana permasalahannya apabila diinginkan salah satu server dibuat supaya dapat menyala terus-menerus. Server yang harus menyala terus-menerus umumnya adalah server yang harus memberi layanan publik misalnya layanan *e-mail*. Dalam kasus ini secara mudah diperlukan UPS yang punya kapasitas tinggi yang dapat mempertahankan daya listrik cukup lama. Persoalan yang muncul adalah bahwa UPS yang punya kapasitas tinggi harganya sangat mahal. UPS dengan kapasitas yang tinggi juga punya keterbatasan

mempertahankan daya listrik sampai waktu tertentu. Setelah UPS tidak dapat lagi mempertahankan daya listrik, maka server harus *shutdown* secara normal juga. Kurang layak diterapkan bahwa pada tiap-tiap server diberi fasilitas UPS yang punya kapasitas tinggi, karena memerlukan pembiayaan yang sangat besar. Harga UPS punya kapasitas tinggi sudah melampaui harga server itu sendiri.

UPS merupakan peralatan yang dapat menyimpan daya listrik sementara sesuai dengan kapasitas yang dimiliki. Dalam menyimpan daya listrik, UPS punya keterbatasan. Berapa pun besarnya kapasitas penyimpanan, suatu ketika pasti akan habis. UPS yang dapat digunakan untuk keperluan pengelolaan daya listrik pada server adalah UPS yang dilengkapi dengan perangkat komunikasi. Perangkat komunikasi dengan server digunakan untuk memberikan isyarat pada server misalnya tentang status sambungan listrik (mati atau hidup) atau status kapasitas penyimpanan listrik (masih berapa persen). Perangkat komunikasi yang dipakai umumnya *port* serial yang menghubungkan UPS dengan server. Isyarat ini penting bagi server untuk melakukan suatu tindakan guna menjaga server dari gangguan atau kerusakan. Misalnya ketika daya listrik terputus, server pengelola segera mematikan server lain yang tidak diperlukan atau ketika simpanan daya listrik sudah hampir habis server harus segera melakukan *shutdown* secara normal.

Dalam sistem jaringan komputer, ada server yang memang harus dipertahankan terus menyala ada juga server yang bisa dimatikan apabila daya listrik padam. Prinsipnya adalah bahwa server yang layak dimatikan boleh

dilakukan proses *shutdown* dengan cara normal, sedangkan server yang harus tetap hidup, harus dipertahankan selama mungkin sesuai dengan kemampuan UPS. Setelah UPS tidak dapat lagi mempertahankan daya listrik, maka server tersebut juga harus dilakukan *shutdown* secara normal juga. Dalam hal ini dapat dipahami bahwa UPS hanya digunakan untuk mempertahankan daya listrik sementara sampai batas waktu kemampuan UPS menyimpan daya listrik. Selama batas waktu tersebut semua server dalam sistem jaringan komputer harus dikelola sedemikian, sehingga dapat terhindar dari kerusakan perangkat keras dan perangkat lunak akibat hilangnya daya listrik secara mendadak.

Dalam penelitian ini dicoba untuk membuat konfigurasi kombinasi bahwa dalam sistem jaringan komputer ada server yang boleh dimatikan dan ada server yang harus tetap hidup. Namun UPS yang dipakai hanya satu tapi punya kapasitas yang tinggi. UPS ini digunakan untuk mempertahankan daya listrik sementara untuk melakukan tindakan pengelolaan daya server pada saat aliran daya listrik mati. Server yang harus tetap menyala diberi tugas untuk mematikan server lain yang boleh dilakukan *shutdown* pada saat sumber daya listrik mati. Untuk mematikan server dari server lain diperlukan perangkat lunak Sshpass. Perangkat lunak Sshpass mengizinkan untuk login pada suatu server secara tidak interaktif. Kemampuan login secara tidak interaktif memungkinkan proses login dapat dikendalikan dari langkah-langkah program. Termasuk dalam hal ini proses apa yang selanjutnya dilakukan setelah proses login berhasil.

Ketika sumber daya listrik sudah mengalir normal lagi, server ini

selanjutnya bertugas menyalakan seluruh server yang ketika listrik padam mengalami *shutdown* secara normal. Proses menghidupkan server harus dilakukan karena apabila server mengalami *shutdown* secara normal, maka server tidak dapat langsung otomatis hidup ketika daya listrik normal kembali. Proses menghidupkan server ini memerlukan perangkat lunak WoL (*Wake on LAN*) serta *motherboard* yang mendukung fitur WoL. Dengan perangkat lunak WoL dimungkinkan untuk menghidupkan suatu server dari keadaan mati melalui jaringan komputer. Perangkat lunak WoL menyalakan server dari keadaan mati dengan cara mengirimkan paket tertentu kepada suatu alamat mesin yang akan dinyalakan.

Dalam penelitian ini diambil asumsi bahwa salah satu server dapat dipertahankan hidup selama mungkin sampai sumber daya listrik normal kembali. Artinya server ini tidak mengalami *shutdown*. Sebab apabila semua server mengalami *shutdown* secara normal, maka server tidak ada yang dapat menyala secara otomatis dan menjalankan prosedur menghidupkan server ketika sumber daya listrik normal kembali. Dengan demikian penggunaan daya listrik dari UPS pada server ini dibuat sedemikian, sehingga menjadi sangat hemat. Salah satu caranya adalah tidak menyalakan monitor. Proses pengaktifan server secara otomatis dari keadaan *shutdown* normal mungkin dapat dilakukan apabila UPS punya fitur tersebut. Dalam penelitian ini, belum digunakan fitur UPS yang dapat menyalakan server dari keadaan *shutdown* normal.

Apabila sumber daya listrik hanya berasal dari satu sumber misalnya PLN,

maka dapat dikatakan kapasitas UPS sebagai batas waktu terakhir server dapat tetap menyala. Namun apabila ada sumber daya listrik selain dari PLN (misalnya dari generator cadangan), maka proses pengelolaan dapat berjalan dengan cara berbeda. Misalnya ketika daya listrik PLN terputus dan selama proses dipertahankan UPS, segera dinyalakan generator cadangan untuk memberi daya listrik pada server. Dalam penelitian ini, hanya dibahas untuk kasus bahwa sumber daya listrik hanya satu. Kasus ini seperti yang terjadi pada laboratorium terpadu STMIK AKAKOM Yogyakarta.

## **B. Tujuan**

Tujuan yang ingin dicapai dari hasil penelitian yang dilakukan adalah sebagai berikut.

- Membuat konfigurasi pemanfaatan UPS untuk menjaga daya yang mencatu server agar tetap menyala.
- Mematikan server secara normal ketika daya listrik mendadak padam, sehingga server dapat menjalani proses *shutdown* secara normal.
- Memanfaatkan teknologi *Wake on LAN* untuk menghidupkan server ketika aliran daya listrik normal.
- Menjaga peralatan server dari kerusakan perangkat lunak dan perangkat keras yang diakibatkan kehilangan daya listrik secara mendadak.

## **C. Manfaat Penelitian**

Manfaat penelitian adalah bahwa hasilnya dapat diterapkan untuk membuat pengelolaan daya listrik UPS secara optimal untuk menjaga agar layanan jaringan yang diberikan server tetap berjalan secara baik. Manfaat lain dari hasil penelitian ini adalah untuk menjaga peralatan server dari kerusakan akibat kehilangan daya listrik secara mendadak.

## BAB II. TINJAUAN PUSTAKA

Penelitian ini melibatkan beberapa materi yang digabungkan untuk menyusun sistem agar dapat bekerja secara baik. Materi yang digabungkan antara lain adalah *Uninterruptible Power Supply* (UPS), *Wake on LAN* (WoL) dan Sshpass. Materi UPS berkaitan dengan pengaturan perangkat lunak yang digunakan untuk memonitor dan mengendalikan perangkat keras. Untuk keperluan ini digunakan perangkat lunak adalah Rups. *Wake on LAN* berkaitan dengan penggunaan perangkat lunak untuk membangkitkan mesin server dari keadaan mati. Untuk keperluan ini digunakan perangkat lunak Ether-Wake. Sshpas berkaitan dengan prosedur mematikan mesin server dari jarak jauh.

Penelitian yang pernah dilakukan berkaitan dengan UPS selama ini hanya berkaitan dengan pengaturan perangkat lunak UPS untuk mengendalikan daya satu komputer. Seperti yang dilakukan Moh. Suryadiman dan kawan kawan dari Badan Tenaga Atom Nasional untuk meneliti pengaruh UPS terhadap kinerja perangkat komputer. Dalam penelitian tersebut dilakukan analisis pengaruh UPS terhadap kinerja perangkat komputer baik *hard disk*, *motherboard* maupun *power supply*. [6]

Kesimpulan dalam penelitian itu adalah kegagalan catu daya sesaat dapat mengakibatkan kerusakan perangkat keras maupun perangkat lunak serta kerugian

waktu dalam hal mengamankan data. Dengan menggunakan UPS dapat dicegah kerusakan perangkat keras dan perangkat lunak akibat dari kegagalan catu daya listrik PLN, kemungkinan *operating system files corrupt*, kemungkinan *data files lost*, kerugian waktu dan besarnya biaya untuk mengembalikan sistem kembali up/normal, berkurangnya loyalitas pelanggan karena sistem sering mati, usia pakai komputer akan lebih singkat dan biaya operasional menjadi besar. Saran yang dituliskan dari hasil penelitian adalah sebaiknya setiap pemakaian komputer harus di pasang perangkat UPS agar komputer tidak cepat rusak.

Saran yang diajukan cukup realistis, namun demikian apabila hal demikian diterapkan pada server, maka biaya yang dibutuhkan untuk investasi sedemikian besar. Biasanya server membutuhkan UPS yang punya daya simpan listrik yang besar, sehingga kemungkinan harga UPS menjadi lebih mahal dibandingkan harga server sendiri. Penggunaan UPS pada server digunakan untuk tetap mempertahankan server agar tetap menyala selama mungkin. Karena yang diinginkan adalah server tidak pernah mati dan selalu melayani permintaan dari pengguna sistem jaringan komputer.

Dengan pertimbangan bahwa harga UPS yang punya kapasitas besar sangat mahal dan tidak semua server harus tetap dipertahankan menyala, perlu dibuat konfigurasi yang mana satu UPS dapat memberi catu daya pada beberapa server. Server yang boleh dimatikan pada saat sumber listrik utama mati, akan dimatikan dengan cara tertentu. Sedangkan server yang harus tetap hidup, akan dipertahankan selama mungkin.

Dalam penelitian ini akan dikembangkan konfigurasi perangkat keras maupun perangkat lunak yang memungkinkan satu UPS dapat dipakai oleh beberapa server. Karena UPS hanya menyediakan satu *port* serial yang bisa dipakai, komunikasi UPS dengan server hanya dapat dilakukan pada salah satu server. Server ini nantinya bertindak sebagai pengelola. Konfigurasi punya kemampuan untuk mematikan server lain dari jarak jauh ketika sumber listrik padam dan menghidupkan server lain dari jarak jauh ketika sumber listrik utama pulih kembali.

## BAB III. TEORI

### A. Server

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. Server didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan atau *network operating system*. Server juga menjalankan perangkat lunak administratif yang mengendalikan akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti berkas (*file*) atau alat pencetak (*printer*), dan memberikan akses kepada *workstation* anggota jaringan.[12]

Pengertian lain server didasarkan pada aplikasi yang diinstal dalam mesin. Perangkat lunak apa yang diinstal pada mesin tersebut, menentukan jenis server. Beberapa layanan aplikasi untuk server antara lain adalah: [12]

- Name server atau DNS server : menyediakan layanan pemetaan IP ke nama atau sebaliknya,
- *Proxy* server : menyediakan layanan *proxy*,
- Web server : menyediakan layanan situs HTML,
- *Mail* Server : menyediakan layanan pengiriman dan penerimaan *e-mail*.

- *File server* : menyediakan layanan berkas, dan
- *Data base server* : menyediakan layanan *data base*.

Dalam komputasi, server nama (*name server*) terdiri dari program atau komputer server yang menerapkan protokol layanan nama. Name server membuat peta apa yang biasa dikenali manusia dengan sistem internal sering numerik atau komponen pemberian alamat.[10]

Yang paling menonjol dari server nama dalam operasi sekarang adalah server nama dari *Domain Name System* (DNS), salah satu dari dua ruang nama utama pada Internet. Fungsi yang paling penting dari server DNS adalah menerjemahkan (resolusi) dari nama domain manusiawi yang mudah diingat dan nama *host* ke Internet numerik yang sesuai alamat *Internet Protocol* (IP). Ruang nama Internet utama yang kedua digunakan untuk mengidentifikasi dan menemukan sistem komputer dan sumber daya di Internet.[10]

Dalam jaringan komputer, server *proxy* adalah server (sistem komputer atau program aplikasi) yang bertindak sebagai perantara permintaan dari klien yang mencari sumber daya dari server lain. Klien terhubung ke server *proxy*, meminta beberapa layanan, seperti file, koneksi, halaman web, atau sumber daya lainnya, yang tersedia dari server yang berbeda. Server *proxy* mengevaluasi permintaan menurut aturan filter. Sebagai contoh, *proxy* mungkin melakukan filter lalu lintas menurut alamat IP atau protokol. Jika permintaan diizinkan oleh filter, *proxy* menyediakan sumber daya dengan menghubungkan ke server yang relevan

dan meminta layanan atas nama klien. Sebuah server *proxy* tambahan dapat mengubah permintaan klien atau tanggapan dari server, dan kadang-kadang mungkin melayani permintaan tanpa menghubungi server yang ditentukan.[11]

Sebuah server *proxy* yang melewati permintaan dan balasan tanpa modifikasi biasanya disebut *gateway* atau kadang-kadang disebut *proxy tunneling*. Server *proxy* dapat ditempatkan pada komputer lokal pengguna atau pada berbagai titik antara pengguna dan server tujuan pada *Internet*. *proxy reverse* (biasanya *proxy* yang berhadapan dengan Internet) digunakan sebagai sebuah *front-end* untuk mengendalikan dan melindungi akses ke server pada jaringan lokal, umumnya juga melakukan tugas-tugas seperti *load-balancing*, autentikasi, deskripsi atau *caching*. [11]

Server web dapat disebut sebagai perangkat keras (komputer) atau perangkat lunak (aplikasi komputer) yang membantu untuk memberikan konten yang dapat diakses melalui Internet. Server web adalah apa yang memungkinkan untuk dapat mengakses konten seperti halaman web atau data lain dari mana saja asalkan terhubung ke Internet. Perangkat keras menyimpan konten, sedangkan perangkat lunak membuat konten dapat diakses melalui Internet. [14]

Penggunaan paling umum dari server web adalah untuk *hosting website*. Namun demikian ada kegunaan lain dari server web seperti penyimpanan data atau untuk menjalankan aplikasi *enterprise*. Banyak cara yang dapat dilakukan untuk meminta konten dari web server. Permintaan paling umum adalah *Hypertext Transfer Protocol* (HTTP), tetapi ada juga permintaan lain seperti

*Internet Message Access Protocol* (IMAP) atau *File Transfer Protocol* (FTP). [14]

File server adalah sebuah komputer yang terpasang ke jaringan yang memiliki tujuan utama menyediakan lokasi untuk akses disk secara berbagi (*shared disc access*), yaitu berbagi penyimpanan file komputer (seperti dokumen, file suara, foto, film, gambar, *data base*, dan lain-lain) yang dapat diakses oleh *workstation* yang melekat pada jaringan komputer. Istilah server menyoroti peran mesin dalam skema *client-server*, yang mana klien adalah *workstation* yang menggunakan penyimpanan. Sebuah file server biasanya tidak melakukan sesuatu perhitungan dan tidak menjalankan program atas nama klien. Server ini dirancang terutama untuk memungkinkan penyimpanan dan pengambilan data cepat yang mana perhitungan berat disediakan oleh *workstation*. [9]

File server biasanya ditemukan di sekolah-sekolah dan kantor-kantor dan jarang terlihat pada penyedia layanan Internet lokal dengan menggunakan LAN untuk menghubungkan komputer klien mereka. [9]. Server *data base* adalah program komputer yang menyediakan layanan *data base* untuk program komputer lain atau komputer yang didefinisikan oleh model *client-server*. Istilah ini juga dapat merujuk pada komputer yang didedikasikan untuk menjalankan program yang sesuai. Sistem manajemen *data base* yang sering menyediakan fungsionalitas *data base* server dan beberapa DBMS (misalnya, MySQL) bergantung secara eksklusif pada model *client-server* untuk mengakses *data base*. [8]

Server tersebut diakses baik melalui suatu "*front-end*" yang berjalan di komputer pengguna yang menampilkan data yang diminta atau "*back-end*" yang berjalan pada server dan menangani tugas-tugas seperti analisis data dan penyimpanan. Dalam model *master-slave*, server *data base* master adalah lokasi pusat dan utama data, sementara server *data base slave* disinkronisasi secara *backup* dari master yang bertindak sebagai *proxy*. [8]

Beberapa contoh dari server basis data adalah Oracle, DB2, Informix, Ingres, SQL Server. Setiap server menggunakan query logika dan struktur sendiri. Bahasa *query* SQL kurang lebih sama di semua server *data base*. [8]

## **B. UPS (*Uninterruptible Power Supply*)**

UPS adalah perangkat yang biasanya menggunakan baterai cadangan sebagai catuan daya alternatif, untuk dapat memberikan catu daya yang tidak terganggu untuk perangkat elektronik yang terpasang. UPS merupakan sistem penyedia daya listrik yang sangat penting dan diperlukan sekaligus dijadikan sebagai sarana terakhir untuk menanggulangi kegagalan daya serta kerusakan sistem dan perangkat keras. UPS merupakan sistem yang sangat penting dan sangat diperlukan pada banyak perusahaan penyedia jasa telekomunikasi, jasa informasi, penyedia jasa Internet dan banyak lagi. Dapat dibayangkan berapa besar kerugian yang timbul akibat kegagalan daya listrik jika sistem tersebut tidak dilindungi dengan UPS.[13]

### Fungsi Utama dari UPS [13]

- Dapat memberikan daya listrik sementara ketika terjadi kegagalan daya pada listrik utama.
- Memberikan kesempatan waktu yang cukup untuk segera menghidupkan generator sebagai pengganti listrik utama.
- Memberikan kesempatan waktu yang cukup untuk segera melakukan back up data dan mengamankan sistem operasi (OS) dengan melakukan *shutdown* sesuai prosedur ketika listrik utama padam.
- Mengamankan sistem komputer dari gangguan-gangguan listrik yang dapat mengganggu sistem komputer baik berupa kerusakan perangkat lunak, data maupun kerusakan perangkat keras.
- UPS secara otomatis dapat melakukan stabilisasi tegangan ketika terjadi perubahan tegangan pada *input*, sehingga tegangan *output* yang digunakan oleh sistem komputer berupa tegangan yang stabil.
- UPS dapat melakukan diagnosis dan pengelolaan terhadap dirinya sendiri, sehingga memudahkan pengguna untuk mengantisipasi jika akan terjadi gangguan terhadap sistem.
- *User friendly* dan mudah dalam instalasi.
- Pengguna dapat melakukan kontrol UPS melalui jaringan LAN dengan menambahkan beberapa aksesori yang diperlukan.
- Dapat diintegrasikan dengan jaringan Internet.
- Notifikasi jika terjadi kegagalan dengan melakukan pengaturan perangkat

lunak pengelolaan UPS.

Jenis-jenis UPS berdasarkan cara kerjanya [13]

- *Line-interactive* UPS

Pada UPS jenis ini diberi tambahan alat AVR (*automatic voltage regulator*) yang berfungsi mengatur tegangan dari suplai daya ke peralatan.

- *On-line* UPS

Pada UPS jenis ini terdapat satu *rectifier* dan satu *inverter* yang terpisah. UPS jenis ini lebih mahal dibandingkan dengan dua jenis UPS lainnya. Dalam keadaan gangguan, suplai daya ke *rectifier* akan diblok, sehingga akan ada arus DC dari baterai ke *inverter* yang kemudian diubah menjadi AC.

- *Off-line* UPS

UPS jenis ini merupakan UPS paling murah diantara jenis UPS yang lain. Karena *rectifier* dan *inverter* berada dalam satu unit. Dalam keadaan gangguan, *switch* akan berpindah, sehingga suplai daya dari catu utama mengalami blok. Akibatnya akan mengalir arus DC dari baterai menuju *inverter*.

Komponen-komponen UPS [13]

- Baterai

Jenis baterai yang digunakan UPS umumnya adalah *lead-acid* atau *nikel-cadmium*. Baterai ini umumnya mampu menjadi sumber tegangan cadangan

maksimal selama 30 menit.

- *Rectifier* (penyearah)

Penyearah berfungsi untuk mengubah arus AC menjadi arus DC dari catu listrik utama. Hal ini bermanfaat pada saat pengisian baterai.

- *Inverter*

Kebalikan dari penyearah, *inverter* berfungsi untuk mengubah arus DC dari baterai menjadi arus AC. Hal ini dilakukan pada saat baterai pada UPS digunakan untuk memberikan tegangan ke komputer.

UPS bekerja berdasar kepekaan tegangan. UPS akan menemukan penyimpangan jalur tegangan (*line voltage*) misalnya, kenaikan tajam, kerendahan, gelombang dan juga penyimpangan yang disebabkan oleh pemakaian dengan alat pembangkit tenaga listrik yang murah. Karena gagal, UPS akan berpindah ke operasi *on-battery* atau baterai hidup sebagai reaksi kepada penyimpangan untuk melindungi bebannya (*load*). Jika kualitas listrik kurang, UPS mungkin akan sering berubah ke operasi *on-battery*. Kalau beban bisa berfungsi dengan baik dalam kondisi tersebut, kapasitas dan umur baterai dapat bertahan lama melalui penurunan kepekaan UPS.

### **C. WoL (*Wake on LAN*)**

*Wake on LAN* adalah standar jaringan komputer Ethernet yang memungkinkan suatu komputer untuk dihidupkan atau dibangun

menggunakan suatu pesan yang dikirim melalui jaringan. Pesan tersebut biasanya dikirim oleh suatu program sederhana yang dijalankan pada komputer lain dalam sistem jaringan area terbatas. *Wake on LAN* kadang-kadang disingkat dengan *WoL* (dengan huruf o kecil). Istilah lain yang setara *Wake on LAN* adalah *Wake On WAN*, *Remote Wake-up*, *Power On By LAN*, *Power Up by LAN*, *Resume by LAN*, *Resume on LAN*, *Wake Up on LAN*. Dalam hal membangunkan komputer yang melakukan komunikasi melalui Wi-Fi, sebuah standar tambahan yang disebut *Wake on Wireless LAN* (WoWLAN) harus digunakan. Standar *WoL* dan *WoWLAN* sering ditambahkan oleh vendor untuk menyediakan protokol-transparan layanan on-demand, misalnya di fitur *Wake on Demand* pada Apple Bonjour.[7]

### **1. Kebutuhan Sistem (*System requirements*)**

Dukungan *Wake on LAN* diterapkan pada *motherboard* (BIOS) dari sebuah komputer dan antarmuka jaringan (*firmware*), dan akibatnya tidak tergantung pada sistem operasi (dan *driver* NIC) yang berjalan pada perangkat keras. Beberapa sistem operasi dapat mengendalikan perilaku *Wake on LAN* melalui driver perangkat keras. Jika antarmuka jaringan adalah *plug-in* kartu bukan yang terintegrasi ke dalam *motherboard*, kartu juga harus terhubung ke *motherboard* dengan kabel. *Motherboard* dengan pengendali (*controller*) Ethernet yang melekat yang mendukung *Wake on LAN* tidak perlu kabel. Catu daya harus memenuhi spesifikasi ATX 2,01.[7]

## 2. Cara Kerja WoL

*Wake on LAN* diterapkan menggunakan pesan jaringan khusus yang disebut paket ajaib (*magic packet*). Paket ajaib berisi alamat MAC dari komputer tujuan. Komputer yang mendengarkan (*listening*) selalu menunggu paket ajaib yang ditujukan kepadanya dan kemudian mulai membangunkan sistem apabila paket tersebut diterima.[7]

Paket ajaib dikirim pada *layer* data *link* atau *layer* 2 pada model OSI dan dilakukan *broadcast* ke seluruh NIC menggunakan alamat *broadcast* jaringan, *layer* *IP-address* (*layer* 3 pada model OSI) tidak digunakan. Inilah sebabnya mengapa *Wake on LAN* adalah *platform-independen*. Setiap aplikasi, pada platform apa pun, bisa membangunkan komputer yang berjalan pada platform lain. Ini adalah kesalahpahaman umum bahwa karena *Wake on LAN* dibangun di atas teknologi *broadcast* hanya dapat digunakan dalam sub jaringan tertentu. Sementara ini umumnya kasus ada beberapa pengecualian.[7]

Agar *Wake on LAN* bekerja, bagian dari antarmuka jaringan harus tetap siaga. Ini mengkonsumsi daya siaga yang jauh lebih sedikit dibanding daya operasi normal. Jika *Wake on LAN* tidak diperlukan, membuat tidak aktif fitur ini dapat mengurangi sedikit konsumsi daya saat komputer dimatikan namun masih dalam kondisi daya terpasang [7]

## 3. Paket Ajaib (*Magic Packet*)

Paket ajaib adalah *frame broadcast* yang mengandung apa saja dalam muatan 6 byte dari semua *frame* (FF FF FF FF FF FF dalam heksadesimal), diikuti dengan pengulangan enam belas dari 48-bit alamat MAC komputer target. Karena paket ajaib hanya dicermati untuk *string* di atas, dan sebenarnya tidak diurai oleh protokol *stack* penuh, mungkin akan dikirim sebagai protokol *layer-network* dan *layer transport*. Hal ini biasanya dikirim sebagai *datagram* UDP ke *port* 7 atau 9, namun sebenarnya dapat dikirim pada semua *port*.<sup>[7]</sup>

### **Keterbatasan Paket Ajaib**

Paket ajaib standar memiliki keterbatasan dasar sebagai berikut:<sup>[7]</sup>

- Membutuhkan komputer tujuan alamat MAC
- Tidak memberikan konfirmasi pengiriman
- Mungkin tidak bekerja di luar jaringan lokal

Alasan utama untuk ini adalah bahwa implementasi *Wake on LAN* ini dirancang sangat sederhana dan cepat diproses oleh sirkuit pada kartu antarmuka jaringan (NIC) dengan kebutuhan daya yang minimal.

### ***Broadcast Terarah Subnet***

Keterbatasan utama dari standar *broadcast Wake on LAN* adalah bahwa paket *broadcast* umumnya tidak diarahkan. Hal ini mencegah teknik WoL untuk digunakan dalam jaringan yang lebih besar atau melalui Internet. *Subnet Directed*

*Broadcasts* (SDB) dapat digunakan untuk mengatasi keterbatasan ini. SDB mungkin memerlukan perubahan konfigurasi *router intermediate*. SDB diperlakukan sebagai paket jaringan normal sampai diproses oleh *router* akhir. *Router* ini mengubah paket ke dalam paket *broadcast* yang benar. Teknik ini memungkinkan *broadcast* yang akan dimulai pada jaringan *remote* tetapi membutuhkan campur tangan semua *router* untuk meneruskan SDB. Ketika menyiapkan jaringan untuk meneruskan paket SDB, perhatian harus diambil untuk menyaring sedemikian, sehingga hanya paket SDB yang diinginkan (misalnya WoL) yang diizinkan, kalau tidak jaringan menjadi tidak terlindungi terhadap serangan DDoS seperti *Smurf Attack*. [7]

### **Pengiriman Paket Ajaib**

Ada banyak cara untuk mengirimkan paket ajaib. Banyak tersedia Perangkat lunak untuk semua platform modern, termasuk Windows, Apple dan Linux. Juga ada situs web di Internet yang memungkinkan sebuah paket ajaib yang akan dikirim secara *on-line* tanpa biaya. Contoh kode sumber untuk pengembang untuk menambahkan *Wake on LAN* untuk program sudah tersedia di banyak bahasa komputer. [7] Beberapa *router* dapat mengirimkan paket ajaib untuk LAN, misalnya *router* Mikrotik.

### **Keamanan Paket Ajaib**

Paket ajaib dikirim melalui *layer data link* atau *layer 2* OSI, yang dapat

digunakan atau disalahgunakan oleh siapa pun di LAN yang sama, kecuali peralatan LAN L2 mampu untuk diatur melakukan penyaringan lalu lintas tersebut untuk mencocokkan persyaratan keamanan situs-lebar (*site-wide*). Firewall dapat digunakan untuk mencegah klien dalam WAN publik dari mengakses alamat *broadcast* dari segmen LAN didalamnya.[7]

NIC tertentu mendukung fitur keamanan yang disebut "*SecureOn*". Hal ini memungkinkan pengguna untuk menyimpan dalam NIC sandi heksadesimal dari 6 byte. Klien harus menambahkan *password* ini untuk paket ajaib. NIC membangunkan sistem hanya jika alamat MAC dan *password* yang benar. Ini tindakan pengamanan yang secara signifikan mengurangi risiko serangan *brute force*, dengan meningkatkan ruang pencarian oleh 48 bit (6 byte), sampai dengan 296 kombinasi jika alamat MAC sepenuhnya diketahui. Penyalahgunaan fitur *Wake on LAN* hanya memungkinkan komputer untuk diaktifkan, tidak dengan sendirinya mengabaikan *password* dan bentuk keamanan lainnya.[7]

Beberapa PC memasukkan teknologi yang dibangun pada *chipset* untuk meningkatkan keamanan *Wake on LAN*. Sebagai contoh, Intel AMT (komponen teknologi Intel vPro), memasukkan *Transport Layer Security* (TLS), sebuah protokol standar-industri yang kuat enkripsi. AMT menggunakan enkripsi TLS untuk mengamankan saluran komunikasi *out-of-band* pada PC *AMT-based* untuk manajemen perintah jarak jauh seperti *Wake on LAN*. AMT mengamankan saluran komunikasi *Advanced Encryption Standard* (AES) 128-bit enkripsi dan RSA kunci dengan panjang modulus 2.048 bit. Karena komunikasi terenkrip adalah

*out-of-band*, perangkat keras PC dan *firmware* menerima paket ajaib sebelum lalu lintas jaringan mencapai *software stack* pada sistem operasi (OS). Karena komunikasi terenkripsi terjadi di bawah level OS, pengiriman paket ajaib kurang rentan terhadap serangan virus, *worm*, dan ancaman lainnya yang biasanya level targetnya OS.[7]

#### **D. Ether-Wake**

Ether-wake adalah perangkat lunak untuk mengirim paket ajaib bagi keperluan *Wake on LAN*. Ether-wake adalah program yang menghasilkan dan mengirim paket ajaib WoL untuk menghidupkan mesin yang telah dimatikan secara *soft-powered-down* (ACPI D3-*warm state*). Perangkat lunak ini menghasilkan format paket ajaib standar AMD, secara opsional dapat mencakup *password*. Satu parameter tunggal yang diperlukan adalah alamat MAC atau ID *host* yang dapat ditranslasi menjadi alamat MAC oleh *data base ether* dalam *nsswitch.conf*. [1]

Sintaksis yang didukung untuk menggunakan perintah ether-wake adalah sebagai berikut. [1]

ether-wake [opsi] Host-ID

Opsi yang diizinkan bagi sintaksis tersebut adalah sebagai berikut.

- -b  
mengirim paket *wake-up* pada alamat *broadcast*
- -D

meningkatkan level Debug

- -i ifname  
menggunakan antarmuka ifname untuk menggantikan antarmuka bawaan eth0
- -p passwd  
memberikan parameter *password*
- -V  
menampilkan informasi versi program

## E. Sshpass

Sshpass adalah perangkat lunak yang dapat digunakan untuk melakukan autentikasi *password* protokol SSH secara tidak interaktif. Dengan demikian Sshpass disebut "*interactive keyboard password authentication*". Hal demikian sebagai pengganti yang umum dilakukan kebanyakan pengguna yaitu menggunakan autentikasi kunci publik yang aman protokol SSH. [4]

Sshpass adalah suatu utilitas yang dirancang untuk menjalankan ssh menggunakan suatu mode untuk menggantikan interaktif *keybooad* dalam melakukan autentikasi *password*. Autentikasi *password* dilakukan secara tidak interaktif. Ssh menggunakan akses TTY secara langsung untuk memastikan kebenaran *password* yang dimasukkan secara interaktif melalui *keyboard* oleh pengguna. Sshpass menjalankan ssh dalam suatu TTY khusus, kemudian membuat trik sedemikian, sehingga seperti pengguna memasukkan *password* secara interaktif [2]

Perintah untuk menjalankan ssh diatur menggunakan opsi yang dimiliki oleh perintah sshpass. Secara khusus perintah sshpass akan seperti ssh beserta argumen-argumennya, tetapi perintah sshpass dilengkapi dengan argumen lain untuk menjalankan perintah. *Prompt password* yang digunakan oleh ssh secara paksa diubah melalui suatu kode program dalam sshpass.

Sshpass memiliki beberapa opsi ketika dijalankan. Jika tidak ada opsi yang diberikan, sshpass akan membaca *password* dari input divais standar. Pengguna diizinkan memberi lebih dari satu pilihan sumber untuk *password*. Opsi-opsi yang dimiliki sshpass supaya dapat menjalankan fungsi login secara tidak interaktif adalah sebagai berikut. [2]

- -p password

Opsi ini digunakan untuk menuliskan *password* pada command-line. Hal demikian perlu dipertimbangkan masalah keamanan, karena artinya *password* dituliskan secara tidak terenkripsi dan dapat dilihat dalam perintahnya.

- -f filename

*Password* dituliskan pada baris pertama dalam suatu file bernama filename. Dalam hal ini, *password* disimpan pada satu file tertentu.

- -d number

Pada opsi ini, number adalah suatu file deskriptor yang diwarisi oleh sshpass dari yang menjalankan. *password* dibaca dari file deskriptor yang dibuka.

- -e

Opsi ini dipakai dengan maksud, agar *password* diambil dari variabel

environment yang diberi nama "SSHPASS"

### 1. Pertimbangan Keamanan (*Security Considerations*) [2]

Pertama dan terpenting, pengguna Sshpass seharusnya menyadari bahwa mengapa ssh hanya mengizinkan untuk mendapatkan *password* secara interaktif bukan tanpa alasan. Sangat hampir tidak mungkin untuk menyimpan dengan aman *password*. Pengguna Sshpass harus mempertimbangkan apakah otentikasi kunci publik ssh memberikan pengalaman pengguna yang sama, sedangkan yang melibatkan lebih sedikit kerumitan, menjadi lebih aman.

Opsi `-p` dipertimbangkan sebagai opsi yang paling aman dari semua opsi sshpass. Semua pengguna sistem dapat melihat *password* pada baris perintah dengan perintah sederhana `"ps"`. Sshpass tidak akan mencoba untuk menyembunyikan *password*, sebagai upaya menciptakan kondisi pacu (*race condition*) tanpa benar-benar memecahkan masalah. Pengguna Sshpass dianjurkan untuk menggunakan salah satu teknik pelewatan *password*, yang semuanya lebih aman.

Sshpass sampai saat ini masih dalam tahapan awal pengembangan. Dengan demikian, terdapatnya bug sangat dimungkinkan. Secara khusus, jika *password* dibaca dari *stdin* (tidak ada opsi *password* sama sekali), ada kemungkinan bahwa beberapa masukan yang bertujuan untuk diteruskan kepada ssh akan dibaca oleh sshpass dan hilang.

Perangkat lunak Sshpass disebarakan dalam bentuk kode sumber (*source*

*code*) dan harus diinstal sesuai dengan distro Linux yang dipakai. Sebelum diinstal, kode sumber harus dikompilasi terlebih dahulu. Proses kompilasi dan instalasi dilakukan dengan memberi perintah `./configure`, `make` dan `make install` dari *shell* pada direktori kerja kode sumber.

Sekrip `configure` mencoba untuk menebak nilai yang benar untuk berbagai variabel yang tergantung sistem (*system-dependent*) yang digunakan selama kompilasi kemudian menggunakan nilai-nilai tersebut untuk membuat file `Makefile` yang ada dalam setiap direktori paket. Hal ini juga dapat digunakan untuk membuat satu atau lebih file `.h` yang mengandung definisi yang tergantung sistem. Akhirnya, sekrip `configure` menciptakan sebuah sekrip `config.status` yang dapat dijalankan pada waktu lain untuk menciptakan kembali konfigurasi terbaru, dan suatu berkas `config.log` yang berisi *output compiler* (berguna terutama untuk *debugging configure*).

Dapat juga digunakan file opsional (biasanya disebut `config.cache` dan diaktifkan dengan `--cache-file = config.cache` atau hanya `-C`) yang menyimpan hasil tes untuk mempercepat diatur ulang. *Caching* tidak diaktifkan secara bawaan untuk mencegah masalah karena penggunaan *cache file* yang lama.

*File* `configure.ac` (atau `configure.in`) digunakan untuk membuat `configure` oleh sebuah program yang disebut `autoconf`. File `configure.ac` diperlukan jika diinginkan untuk mengubah atau memperbarui `configure` menggunakan versi yang lebih baru dari `autoconf`.

Langkah paling mudah untuk kompilasi paket `sshpas` adalah: [3]

- cd ke direktori yang berisi paket dan tipe kode sumber
- jalankan ./configure untuk mengatur paket pada sistem.
- Menjalankan configure mungkin memakan waktu. Sementara berjalan, configure mencetak beberapa pesan yang menampilkan fitur yang diperiksa.
- Ketik make untuk kompilasi paket.
- Opsional, diketikkan make check untuk menjalankan tes terhadap paket.
- Ketik make install untuk menginstal program, file data dan dokumentasi.
- Program biner dan file objek dapat dihapus dari direktori kode sumber dengan mengetikkan make clean. Untuk membuang file yang dibuat configure (sehingga dapat melakukan kompilasi paket dari komputer yang berbeda), diketikkan make distclean. Ada juga target make maintainer-clean, tapi itu dimaksudkan terutama untuk pengembang paket. Jika digunakan target tersebut, mungkin diperlukan banyak program lain dalam rangka untuk mengembangkan file yang disertakan pada distribusi.
- Seringkali, juga diketikkan make uninstall untuk menghapus file yang diinstal.

## **F. Rups**

Rups adalah utilitas perangkat lunak untuk memonitor peralatan UPS. Rups dapat mendeteksi status UPS cerdas yang ditempatkan secara lokal, melalui kabel yang dihubungkan melalui *port* serial pada sistem. Selain itu, Rups dapat digunakan untuk memonitor status UPS yang terletak pada jarak jauh menggunakan protokol SNMP melalui komunikasi jaringan komputer. Bila terjadi

kegagalan daya listrik, Rups akan mengambil isyarat kegagalan daya dan mengirimkan pesan peringatan ke seluruh pengguna sistem. Setelah pewaktu *shutdown* berhitung mundur sampai nol, Rups akan menjalankan prosedur *shutdown*, atau pengguna dapat mengatur prosedur *shutdown* sendiri. Setelah prosedur *shutdown* dijalankan, Rups akan melaksanakan *shutdown* untuk sistem operasi dan selanjutnya mematikan UPS. Setelah daya listrik pulih kembali, UPS akan aktif dan melakukan boot sistem. [5]

### 1. Fitur-Fitur

Fitur-fitur yang dimiliki oleh Rups adalah sebagai berikut. [5]

- deteksi jalur kegagalan dan baterai rendah.
- perekaman dan penyebaran kondisi abnormal power
- monitoring status power melalui sistem *port* serial RS232 (tidak memerlukan perangkat keras tambahan)
- delay time *shutdown* secara terprogram.
- pelaksanaan tugas secara *batch* sebelum sistem *shutdown*.
- fungsi UPS *turn-off* dan fungsi *night-off* yang dapat diprogram pengguna.
- memonitor UPS dari jauh menggunakan agen SNMP pada sistem jaringan. mengikuti UPS IETF MIB (RFC 1628) atau informasi *proprietary* UPS MIB.
- DC power input yang terisolasi untuk melindungi rangkaian antarmuka UPS dari gejolak *host* RS232 (hanya dalam Rups untuk plus Unix).

## 2. Sistem Operasi yang Didukung

Rups untuk Unix dapat menyediakan layanan UPS pada sistem operasi UNIX berikut yang memiliki *port* serial RS232 khusus. [5]

- SCO UNIX v3.x, v4.x, v5.x; XENIX v2.x; UnixWare v2.x
- DEC OSF / 1 v2.x, v3.x; \* ULTRIX v4.x; Digital UNIX v4.x
- \* IBM AIX v3.x, v4.x
- \* NEC UP / EWS 4800 SVR4 ver 4.x
- HP HP-UX v8.x, v9.x, v10.x; Apollo UX & VUE
- SGI IRIX v6.x
- Sun SunOS 4.x ver (SPARC, ULTRA)
- Sun Solaris ver 2.x (SPARC, ULTRA, x86)
- Linux ver 2.x
- \* Motorola 4.x ver SVR4; \* AIX Motorola ver 4.x
- FreeBSD ver 2.x

\*hanya didukung dalam Rups untuk versi Unix \* plus.

## 3. Instalasi Rups

Instalasi perangkat lunak Rups dapat dilakukan dengan mengikuti langkah-langkah sebagai berikut. [5]

- login sebagai root.
- masukkan CD pada CD-ROM Driver.
  - ✓ mount CD dalam sistem direktori file Unix misalnya /cdrom.

pada sistem Linux, perintahnya adalah sebagai berikut.

```
# mount -t iso9660 /dev/hdd /mnt/cdrom
```

(nama divais yang digunakan tergantung sistem)

- ✓ file target disalin, diberi nama oleh sistem target, di bawah sub direktori /cdrom ke dalam direktori target /tmp.

```
# cp /mnt/cdrom/LINUX/linux.tar.gz /tmp
```

- jalankan program instalasi:

```
# cd /tmp
```

```
# uncompress linux.tar.gz
```

```
# tar xvf linux.tar.gz
```

```
# cd Rups (or rups1)
```

```
# ./install
```

#### 4. Konfigurasi Rups

Setelah instalasi Rups selesai, Rups harus dikonfigurasi untuk memantau daya sistem. Masukkan perintah di bawah ini untuk memulai program konfigurasi Rups yang akan menampilkan dialog konfigurasi

```
# cd /etc/RUPS
```

```
# /rupsc config
```

Pilih opsi dengan memasukkan huruf pada awal tiap item untuk mengubah konfigurasi saat ini. Setelah konfigurasi dimodifikasi, tekan S untuk menyimpan konfigurasi yang baru atau tekan Q untuk berhenti dari konfigurasi Rups.

Beberapa konfigurasi penting pada Rups adalah sebagai berikut.

- Konfigurasi *Shutdown* Sistem

Tugas yang paling penting dari Rups adalah untuk mematikan sistem dengan aman bila listrik mati.

- Konfigurasi *port* Serial

Rups menyediakan fungsi lokal dan *remote monitoring* UPS. Ketika UPS terletak di samping komputer *host*, Rups mendeteksi status daya dari *port* serial *host*. Silakan ikuti petunjuk di bawah ini untuk mengkonfigurasi *port* serial komputer *host*.

- *Monitoring* UPS Jarak Jauh

Rups dapat digunakan untuk memonitor agen UPS *remote* SNMP. Selain itu, juga harus periksa pengaturan keamanan dari agen SNMP untuk memastikan informasi UPS dapat diakses oleh *host* komputer ini. Rups membutuhkan hak istimewa membaca untuk melakukan fungsi pemantauan status UPS. Hak istimewa tulis akan diperlukan jika berniat untuk mematikan daya UPS setelah sistem dimatikan. Bila mengkonfigurasi Rups untuk memantau agen SNMP terpencil, beberapa fungsi Rups akan tergantung pada fungsi agen SNMP menyediakan.

### **Start dan Stop Proses Rups**

Setelah instalasi selesai, Rups akan proses latar belakang setiap *startup* sistem waktu. Jika konfigurasi telah dimodifikasi, sistem dapat *direboot* atau *direstart* untuk memperbarui konfigurasi baru. Dapat dipilih [M] pada menu

konfigurasi Rups atau menggunakan perintah di bawah ini untuk memulai Rups secara manual.

```
/etc/RUPS start /rupsc
```

Sebelum menjalankan Rups, pastikan proses *getty* tidak diaktifkan untuk *port* serial yang ditunjuk dan tidak ada proses lain akan menggunakan *port* serial yang sama. Jika proses *getty* telah proses diaktifkan atau lainnya adalah dengan menggunakan *port* serial yang sama untuk komunikasi, Rups tidak akan dapat mendeteksi sinyal dari UPS. Dapat digunakan perintah *ps* untuk memeriksa apakah ada proses lain yang menggunakan *port* TTY yang sama untuk komunikasi serial.

### ***Shutdown Batch Job***

Rups menyediakan fungsi *batch* pekerjaan yang memungkinkan untuk dilakukan berdasarkan proses yang dibuat sebelum *shutdown* kegagalan daya. File perintah adalah file teks biasa. Dapat digunakan editor untuk menambah atau mengubah perintah dalam file ini. Nama file dari file perintah adalah:

```
/etc/RUPS/preshut.bat
```

### ***Shutdown Command File***

File perintah *shutdown* berisi perintah untuk *shutdown* sistem. Perintah ini akan dieksekusi untuk *shutdown* sistem setelah power gagal mundur. Nama file dari file perintah adalah:

`/etc/RUPS/shutdwon.ini`

Dapat dilihat manual operasional sistem UNIX atau ke bagian yang sesuai sebelum melakukan modifikasi apa pun. Pastikan perintah dapat digunakan untuk mematikan sistem setelah memodifikasi *file* ini.

### Informasi Sejarah

Rups mencatat semua kondisi abnormal dalam file log. Bisa digunakan editor teks untuk meninjau informasi ini dan memeriksa apakah ada kondisi listrik mati terjadi. Pesan yang digunakan dalam file log adalah:

```
AC power failure and restore messages.
@@ Thu Jun 15 12:52:04 1995 AC Power Fail @@
## Thu Jun 15 12:52:22 1995 AC Power Restored ##
UPS battery low and recovery messages when AC power failed.
** Thu Jun 15 12:52:20 1995 UPS Battery Low,
Server will be shutdown in 1 min **
** Fri Jun 16 13:49:35 1995 UPS Battery Low
Recovered **
UPS battery low and recovery messages when AC power normal.
** Thu Jun 15 12:52:30 1995 UPS Battery Weak **
** Thu Jun 15 12:52:36 1995 Battery Weak Back to
Normal **
```

## **BAB IV. METODOLOGI**

### **A. Bahan Penelitian**

Bahan materi yang digunakan dalam penelitian berupa buku teks maupun makalah-makalah yang berhubungan dengan teknologi WoL, UPS, jaringan komputer, Rups dan sshpass.

### **B. Alat**

Alat yang digunakan dalam penelitian ini berupa perangkat keras dan perangkat lunak. Perangkat lunak yang digunakan dalam penelitian adalah sebagai berikut.

- Sistem Operasi Linux
- Ether-Wake
- Rups
- Sshpass

Perangkat keras yang digunakan dalam penelitian umumnya adalah perangkat keras jaringan komputer yang meliputi sebagai berikut.

- Server (server gwlabtga.lab.akakom)

- Server laboratorium yang mendukung WoL
- *workstation* yang mendukung WoL
- Switch
- LAN tester
- UPS
- Kabel UTP
- Konektor RJ-45

### C. Jalan Penelitian

Penelitian dilakukan dengan cara mengatur konfigurasi perangkat keras seperti pada Gambar 1. Selanjutnya dilakukan pengaturan perangkat lunak pada server. Server pengelola bertugas untuk melaksanakan pengelolaan pada server lain berkaitan dengan penggunaan sumber daya listrik dari UPS.



Gambar 1. Konfigurasi Perangkat Keras

Sumber daya UPS dipakai untuk menghidupkan seluruh server baik server pengelola dan server-server lain. Hubungan UPS dengan server pengelola, selain untuk mencatu daya listrik juga hubungan untuk memonitor status UPS. Hubungan catu daya antara server pengelola dengan UPS dilakukan melalui jaringan listrik, sedangkan hubungan untuk mengamati status UPS oleh server pengelola dilakukan melalui komunikasi *port* serial RS 232. Hubungan UPS dengan server-server lain berupa hubungan untuk catu daya melalui jaringan listrik. Hubungan antara server-server lain dan server pengelola dilakukan melalui sistem jaringan komputer.

Komputer server pengelola dikonfigurasi supaya dapat memonitor status UPS, melakukan *shutdown* melalui sistem jaringan komputer dan menghidupkan server lain dari keadaan *shutdown* normal. Pada server pengelola harus dipasang perangkat lunak Rups, sshpass dan WoL. Server-server lain harus menyediakan layanan login dan memberi izin masuk superuser secara langsung. Server lain tersebut harus dipasang perangkat lunak server SSH.

UPS digunakan untuk mempertahankan daya listrik sementara, sehingga server dapat melakukan tindakan pengelolaan server pada saat aliran daya listrik mati. Server pengelola harus tetap menyala dan diberi tugas untuk mematikan server lain yang boleh *dishutdown* pada saat sumber daya listrik mati. Untuk mematikan server dari server lain diperlukan perangkat lunak sshpass. Perangkat lunak sshpass mengizinkan untuk login pada suatu server tidak secara interaktif. Ketika sumber daya listrik sudah mengalir normal lagi, server pengelola bertugas

menyalakan seluruh server yang ketika listrik padam *dishutdown* secara normal.

Proses menghidupkan server ini memerlukan perangkat lunak WoL.

Langkah-langkah yang dilakukan pada penelitian ini dapat digambarkan sebagai berikut.

1. konfigurasi perangkat keras seperti pada Gambar 3.1 yang meliputi hubungan antara UPS dengan server pengelola melalui *port* serial RS-232, hubungan antara UPS dengan seluruh server melalui jaringan listrik dan hubungan antar server melalui sistem jaringan komputer,
2. instalasi dan konfigurasi perangkat lunak Rups pada server pengelola,
3. instalasi dan konfigurasi perangkat lunak sshpass pada server pengelola,
4. instalasi dan konfigurasi perangkat lunak WoL pada server pengelola,
5. instalasi dan konfigurasi perangkat lunak ssh server pada seluruh server,
6. pengoperasian sistem untuk jangka waktu tertentu,
7. pengamatan terhadap file rupslog,
8. pembuatan program untuk pengamatan file rupslog,
9. percobaan dengan mematikan jalur sumber listrik utama,
10. pengamatan terhadap hasil serta dibahas, dan
11. pembuatan kesimpulan.

## BAB V. ANALISIS DAN PEMBAHASAN

Penelitian ini dimulai dengan konfigurasi perangkat keras. Konfigurasi perangkat keras dimaksudkan untuk menghubungkan antara satu peralatan dengan peralatan. Ada beberapa hubungan yang perlu dijelaskan sehubungan konfigurasi perangkat keras. Hubungan-hubungan tersebut adalah sebagai berikut.

- Hubungan pertama adalah hubungan antara UPS dengan seluruh server termasuk server pengelola berupa hubungan catu daya melalui jaringan listrik. Selain itu, UPS juga harus memberi catu daya pada peralatan switch yang menghubungkan antar server melalui sistem jaringan komputer.
- Hubungan kedua adalah hubungan antara server pengelola dengan UPS untuk memonitor aktivitas UPS oleh server melalui *port* serial RS-232.
- Hubungan ketiga adalah hubungan antar server untuk komunikasi data melalui sistem jaringan komputer. Hubungan ini dilakukan melalui switch yang juga harus mendapat catu daya dari UPS.

Satu syarat supaya sistem yang dirancang ini bekerja adalah dukungan fitur WoL pada *mainboard*. Pada umumnya dukungan WoL pada *mainboard* harus diaktifkan dari BIOS. Beberapa *mainboard* mengatur dukungan fitur WoL pada menu BIOS yang berbeda-beda, namun umumnya pada bagian yang

berkaitan dengan pengelolaan daya (*power management*). Salah satu pengaturan yang pernah dilakukan adalah dengan cara masuk pada sistem BIOS, kemudian masuk pada menu Power Management Setup, kemudian pada pilihan PME Event Wake-Up dibuat menjadi Enable. Pengaturan ini dilakukan pada seluruh server kecuali pada server pengelola. Fitur WoL pada server pengelola tidak perlu ada atau tidak perlu diaktifkan.

Setelah seluruh perangkat keras dikonfigurasi, langkah selanjutnya adalah konfigurasi perangkat lunak pada masing-masing peralatan. Pada server pengelola dipasang perangkat lunak Sshpass, WoL dan Rups. Sshpass digunakan untuk masuk pada suatu server tanpa menjalankan login interaktif, sehingga login dapat dilakukan secara terprogram. Dengan Sshpass nantinya dapat dilakukan *shutdown* server dari server lain. Untuk keperluan ini tidak dapat dilakukan dengan perangkat lunak Ssh karena proses login harus dilakukan secara terprogram (tidak secara interaktif). Perangkat lunak Ssh memerlukan login secara interaktif. Selanjutnya ketika sudah berhasil login, dilakukan *shutdown* server secara terprogram juga. WoL digunakan untuk mengirim isyarat paket ajaib kepada seluruh server yang berfungsi untuk menghidupkan server dari keadaan mati. Rups digunakan untuk memonitor aktivitas UPS, menangkap isyarat dari UPS dan menjalankan aksi-aksi tertentu berkaitan dengan isyarat yang diterima tersebut. Pada seluruh server dipasang perangkat lunak ssh server. Perangkat lunak ini berfungsi untuk melayani hubungan melalui protokol SSH. Dengan adanya perangkat lunak ini, maka seluruh server dapat dihubungi menggunakan protokol SSH termasuk

menggunakan perangkat lunak Sshpass. Dalam kaitan ini, server Ssh harus memberi izin server lain untuk login root secara langsung.

Aktivitas UPS baik pada saat berjalan secara normal, mengalami pemutusan hubungan dengan sumber listrik maupun pada saat sumber listrik pulih kembali akan diamati oleh perangkat lunak Rups. Suatu isyarat akan dikirim melalui *port* serial komputer apabila UPS mengalami salah satu dari tiga kejadian tersebut. Berdasarkan isyarat yang dikirimkan UPS, Rups menjalankan aksi-aksi tertentu. Aktivitas Rups dicatat dalam suatu file log bernama rupslog yang disimpan dalam direktori `/var/log`.

Berikut ini hasil pengamatan terhadap file rupslog yang merupakan file log dari perangkat lunak Rups. File ini mencatat semua aktivitas Rups yang berkaitan dengan kerja UPS

```

@@ Sun Sep 26 18:14:53 2010 AC Power Fail @@
## Sun Sep 26 18:22:52 2010 AC Power Restored ##
## Sun Sep 26 18:22:52 2010 AC Power Restored ##
@@ Sun Sep 26 19:43:55 2010 AC Power Fail @@
!! Sun Sep 26 19:53:58 2010 System Shutdown Begin !!
@@ Tue Sep 28 17:58:46 2010 AC Power Fail @@
!! Tue Sep 28 18:08:49 2010 System Shutdown Begin !!
@@ Mon Oct 4 18:32:06 2010 AC Power Fail @@
## Mon Oct 4 18:32:20 2010 AC Power Restored ##
## Mon Oct 4 18:32:20 2010 AC Power Restored ##
@@ Mon Oct 4 18:32:22 2010 AC Power Fail @@
!! Mon Oct 4 18:42:25 2010 System Shutdown Begin !!
@@ Mon Oct 11 03:20:37 2010 AC Power Fail @@
!! Mon Oct 11 03:23:40 2010 System Shutdown Begin !!
@@ Thu Oct 28 16:11:21 2010 AC Power Fail @@
!! Thu Oct 28 16:21:25 2010 System Shutdown Begin !!
@@ Thu Nov 4 20:34:40 2010 AC Power Fail @@
@@ Thu Nov 11 15:54:13 2010 AC Power Fail @@
!! Thu Nov 11 16:04:17 2010 System Shutdown Begin !!
@@ Thu Nov 11 22:35:32 2010 AC Power Fail @@
## Thu Nov 11 22:35:47 2010 AC Power Restored ##
## Thu Nov 11 22:35:47 2010 AC Power Restored ##
@@ Thu Nov 11 22:35:48 2010 AC Power Fail @@
!! Thu Nov 11 22:45:51 2010 System Shutdown Begin !!

```

```

@@ Fri Nov 12 22:21:43 2010 AC Power Fail @@
!! Fri Nov 12 22:31:46 2010 System Shutdown Begin !!
@@ Wed Nov 17 10:40:58 2010 AC Power Fail @@
## Wed Nov 17 10:41:13 2010 AC Power Restored ##
## Wed Nov 17 10:41:13 2010 AC Power Restored ##
@@ Wed Nov 17 10:41:24 2010 AC Power Fail @@
## Wed Nov 17 10:41:39 2010 AC Power Restored ##
## Wed Nov 17 10:41:39 2010 AC Power Restored ##
@@ Wed Nov 17 10:41:45 2010 AC Power Fail @@
!! Wed Nov 17 10:51:48 2010 System Shutdown Begin !!

```

Ada tiga kejadian utama yang tercatat pada rupslog berkaitan dengan aktivitas UPS yaitu sebagai berikut.

- Pada saat mengalami kegagalan sumber daya listrik (sumber listrik utama mati), UPS akan mengirim isyarat yang selanjutnya ditulis pada rupslog dan dicatat sebagai AC Power Fail.
- Pada saat sumber listrik utama pulih kembali, UPS juga mengirim isyarat yang akan dicatat pada rupslog dan dicatat sebagai AC Power Restored.
- Pada saat sumber listrik utama mati, salah satu tindakan yang dapat diatur dari perangkat lunak Rups adalah mematikan sistem server setelah beberapa saat sesuai dengan pengaturan yang dilakukan. Kejadian ini dicatat sebagai System Shutdown Begin.

Apabila sumber listrik utama mati dalam waktu yang masih dapat ditangani oleh UPS dalam arti baterai UPS masih dapat mencatu daya pada server, maka Rups masih memonitor status UPS. Apabila ada isyarat dari UPS bahwa listrik utama menyala kembali, maka keadaan akan pulih seperti semula. Apabila sampai batas waktu, sumber listrik utama belum pulih, maka Rups akan

melakukan tindakan mematikan server secara normal.

Proses untuk mematikan server secara normal akan dilakukan menurut pengaturan yang telah ditetapkan dalam file Rups.ini yang diletakkan dalam direktori /etc/rups. File rups.ini akan diciptakan/diperbarui pada saat konfigurasi rups menggunakan perangkat lunak rupsc. Isi file rups.ini adalah sebagai berikut.

```
# This file is generated by 'rupsc'
# Do not edit it by hand

acfail_delay      = 10
battlow_delay     = 1
turnoff_delay     = 30
nightoff_delay    = 3
night_start       = 0
night_end         = 6
countdown         = 0
startup_delay     = 0
master            = 1
local_ups         = 1
local_tty         = /dev/ttyS0
community         = public
agent_ip          = 127.0.0.1
```

Proses mematikan server setelah sumber daya listrik utama mengalami kegagalan dapat diatur berdasarkan waktu tunda dengan menentukan `acfail_delay` atau bisa diatur beberapa saat setelah kondisi baterai UPS low dengan menentukan `battlow_delay`. Yang perlu diingat, penundaan dalam mematikan server setelah baterai dalam kondisi low maksimum hanya sampai 2 menit. Rups juga punya kemampuan untuk mengatur proses mematikan UPS setelah sistem server dimatikan dengan menentukan `turnoff_delay`. Dalam hal ini, setelah server dimatikan, UPS akan menghidupkan pewaktu pada UPS untuk mematikan UPS sendiri. Proses untuk mematikan server berdasar waktu tunda setelah sumber

listrik utama mati (`acfail_delay`) atau setelah baterai dalam kondisi low (`battlow_delay`) tergantung mana yang lebih cepat tercapai.

Yang diinginkan dalam penelitian ini adalah bahwa server pengelola harus dibuat menyala selama mungkin sampai batas waktu kemampuan UPS memberi catu daya. Oleh sebab itu, pengaturan yang paling diperlukan adalah mengatur penundaan waktu mematikan server sebesar mungkin, sehingga batas baterai kondisi low dapat tercapai. Selanjutnya setelah selang waktu tertentu sistem server dimatikan. Pada penelitian ini diatur agar Rups akan mematikan sistem server 1 menit setelah baterai UPS dalam keadaan low. Setelah itu dalam waktu 30 menit UPS akan mati secara otomatis.

Dengan pengaturan bawaan sesuai pengaturan Rups, tujuan penelitian belum bisa tercapai karena yang diinginkan dalam penelitian ini adalah server pengelola dibuat menyala selama mungkin (sampai batas kemampuan UPS memberi catu daya listrik). Pada saat sumber listrik utama mengalami mati, server lain harus dimatikan untuk menghemat sumber daya dari UPS sedemikian, sehingga yang masih beroperasi hanya server pengelola. Lama tidaknya server pengelola beroperasi tergantung pada kapasitas UPS, apabila sampai batas kondisi low UPS sumber listrik utama belum menyala, server pengelola juga akan dimatikan. Dalam penelitian ini diatur nilai `acfail_delay` sebesar 180 menit, `battlow_delay` sebesar 1 menit, dan `turnoff_delay` sebesar 30 menit.

Server lain selain server pengelola harus dimatikan pada saat sumber listrik utama mengalami kegagalan. Proses untuk mematikan server lain tidak

dapat memanfaatkan sekrip *shutdown* bawaan Rups yaitu `/etc/RUPS/preshut.bat`, karena sekrip ini dijalankan pada saat server akan dimatikan oleh Rups. Yang diinginkan adalah, sekrip harus dijalankan pada saat sumber listrik utama mengalami kegagalan. Untuk mengetahui kapan sumber listrik utama mengalami mati atau pun saat pulih kembali, dapat dibaca dari file `rupslog`. Dengan demikian perlu dibuat sekrip yang dapat mengamati isi dari `rupslog` tersebut.

Sekrip untuk pengamatan terhadap file `rupslog` harus berjalan terus-menerus atau setiap selang waktu tertentu harus dikerjakan. Untuk keperluan tersebut, program dapat dituliskan dalam program `crontab`. Program yang dibuat untuk mengamati file `rupslog` ini mempunyai dua macam pekerjaan utama.

- Pada saat sumber listrik utama mengalami mati, yang dilakukan server pengelola adalah mematikan server lain. Proses untuk mematikan server lain dilakukan dengan cara masuk server tersebut menggunakan perintah `sshpass` dan selanjutnya diberikan perintah untuk mematikan server.
- Pada saat sumber listrik utama pulih kembali, server pengelola menghidupkan server lain dari keadaan mati. Proses untuk menghidupkan server lain dengan cara mengirim paket ajaib melalui sistem jaringan komputer. Untuk mengirim paket ajaib ini digunakan perangkat lunak `WoL`.

Berikut ini adalah program untuk mengamati file `rupslog` dan menjalankan perintah untuk mematikan atau pun menghidupkan server dari server pengelola. Sekrip untuk menghidupkan dan mematikan server diberi nama `servercontrol.sh`.

```

#!/bin/sh
# nama sekrip: servercontrol.sh
# sekrip bash untuk mematikan dan menghidupkan server
# dari server pengelola
# e-mail: wagito@yahoo.com

# catatan awal
log="/var/log/servercontrol.log"
log_akhir="/var/log/status_akhir"
log_Rups="/var/log/rupslog"

# mencatat waktu dijalankan
tanggal=date
echo "menjalankan sekrip pada $tanggal" >> $log

#semua server didaftar pada bagian ini
ip_proxy="192.168.11.154"
pass_proxy="pass"
mac_proxy="00:00:00:00:00:00"

#memeriksa status akhir
ckakhir=`tail -1 $log_akhir | grep "on" -c`
echo "status akhir $ckakhir" >> $log

#memeriksa status kegagalan UPS
ckfail=`tail -1 $log_Rups | grep "AC Power Fail" -c`
echo "status gagal $ckfail" >>$log

if [ $ckakhir = 1 ] && [ $ckfail = 1 ]; then
  # prosedur mematikan server dari server pengelola
  ckserver=`ping -c 4 $ip_proxy | grep "64 byte" -c`
  echo "hasil ping sebelum mematikan $ckserver" >>$log

  # jika server hidup, maka perlu dimatikan
  if [ $ckserver > 0 ]; then
    echo "matikan server $ip_proxy" >>$log
    sshpass -p $pass_proxy ssh root@$ip_proxy "cd; /sbin/halt"
  fi
  # dan seterusnya untuk server server lain
  . . . . .
  # mencatat status akhir
  echo "off" > $log_akhir
fi

#memeriksa status restore UPS
ckrestore=`tail -1 $log_Rups | grep "AC Power Restored" -c`
echo "status restore $ckrestore" >>$log

if [ $ckakhir = 0 ] && [ $ckrestore = 1 ]; then
  # prosedur menghidupkan server dari server pengelola
  ckserver=`ping -c 4 $ip_proxy | grep "64 byte" -c`
  echo "hasil ping sebelum menghidupkan $ckserver" >>$log

  # jika server mati, maka perlu dihidupkan
  if [ $check_server = 0 ]; then

```

```

        echo "menghidupkan server $ip_proxy" >>$log
        /sbin/ether-wake -b $mac_proxy
    fi
    # dan seterusnya untuk server server lain
    . . . . .
    # mencatat status akhir
    echo "on" > $log_akhir
fi

```

Beberapa bagian program sengaja disembunyikan atau diganti untuk keamanan sistem sesungguhnya misalnya *password*. Program terdiri dari dua bagian. Bagian pertama adalah untuk mematikan server dari jarak jauh dan bagian kedua untuk menghidupkan server dari jarak jauh. Mematikan server dari jarak jauh dilakukan apabila sumber listrik utama mengalami kegagalan. Untuk menghidupkan server harus diperiksa terlebih dahulu apakah server dalam keadaan hidup. Kalau server sudah dalam keadaan mati, maka proses mematikan server tidak perlu dilakukan. Proses menghidupkan server dilakukan apabila sumber listrik utama sudah pulih kembali. Untuk menghidupkan server harus diperiksa apakah server dalam keadaan mati. Apabila server sudah hidup, maka proses menghidupkan server tidak perlu dilakukan.

Secara umum langkah yang dilakukan apabila sumber listrik utama mengalami kegagalan adalah sebagai berikut.

1. diperiksa status akhir sumber listrik utama
2. diperiksa apakah sumber listrik utama mengalami kegagalan dengan memeriksa status pada file rupslog
3. jika status akhir sumber listrik utama menyala dan sumber listrik utama sekarang mengalami kegagalan, maka dilakukan aksi sebagai berikut

4. diperiksa apakah server dalam keadaan hidup menggunakan perintah ping
5. jika ya, maka dilakukan aksi berikut.
6. masuk pada server menggunakan sshpass
7. server dimatikan dengan perintah halt

Secara umum langkah yang dilakukan apabila sumber listrik utama pulih kembali adalah sebagai berikut.

1. diperiksa status akhir sumber listrik utama
2. diperiksa apakah sumber listrik utama sudah pulih kembali dengan memeriksa status pada file rupslog
3. jika status akhir sumber listrik utama mati dan sumber listrik utama sekarang pulih kembali, maka dilakukan aksi sebagai berikut
4. diperiksa apakah server dalam keadaan hidup menggunakan perintah ping
5. jika tidak, maka dilakukan aksi berikut.
6. server dihidupkan dengan perintah ether-wake

Sekrip program tersebut harus dijalankan setiap waktu. Dalam penelitian ini, sekrip program dijalankan setiap satu menit. Artinya setiap satu menit, diperiksa status akhir sumber listrik utama dan status sumber listrik utama paling baru. Untuk pengaturan supaya sekrip program dapat dijalankan setiap satu menit, perangkat lunak crontab diperlukan. Pengaturan crontab diletakkan pada file `/var/spool/cron/root` yang isinya adalah sebagai berikut.

```
# menjalankan sekrip setiap satu menit  
* * * * * /sbin/servercontrol.sh
```

Secara umum penelitian dapat berjalan secara baik. Perintah-perintah yang terdapat pada sekrip `servercontrol.sh` dapat dijalankan secara betul. Namun demikian, salah satu kelemahan utama pada sistem ini adalah apabila sumber listrik utama mengalami kegagalan dalam waktu yang terlalu lama. Pengertian terlalu lama adalah batas waktu kapasitas UPS masih dapat memberi catu pada server pengelola.

Pada penelitian ini, server pengelola berlaku sebagai pengelola penggunaan daya dari UPS. Pengelolaan ini dilakukan sampai batas waktu daya pada UPS habis setelah beberapa waktu sumber daya utama mengalami kegagalan. Apabila sampai batas waktu tertentu sumber listrik utama tidak mengalami pemulihan, akhirnya server pengelola juga harus dimatikan secara normal. Kondisi ini bisa terjadi apabila sumber listrik utama mengalami kegagalan terlalu lama. Apabila kondisi ini terjadi, maka server pengelola tidak akan dapat menyala secara otomatis pada saat sumber listrik utama pulih kembali. Hal demikian disebabkan karena apabila suatu komputer dimatikan secara normal, maka harus dinyalakan secara manual. Karena tidak dapat menyala secara otomatis, maka tentu saja server pengelola tidak dapat menyalakan server-server lain dengan teknik WoL. Apabila server pengelola sampai kehabisan daya UPS, maka semua sistem akan mati. Untuk memulihkan sistem bekerja dilakukan

secara manual.

Kelemahan konfigurasi ini dapat meminimalkan dengan cara menggunakan server pengelola yang hemat sumber daya. Pemilihan server pengelola yang hemat sumber daya menjadikannya dapat menyala jauh lebih lama. Apabila server pengelola dapat menyala lebih lama dengan daya UPS diharapkan ketika sumber daya utama pulih kembali server pengelola masih menyala. Apabila masih menyala ketika sumber listrik utama pulih, maka dapat dideteksi oleh server pengelola dan selanjutnya dapat digunakan untuk menyalakan server lain dengan teknik WoL.

## **BAB VI. KESIMPULAN DAN SARAN**

### **A. Kesimpulan**

Beberapa kesimpulan yang dapat ditarik dari pembahasan tentang penelitian ini adalah sebagai berikut.

- Secara umum sistem dapat bekerja secara baik, dapat menangkap isyarat yang dikirim oleh UPS.
- Perangkat lunak Sshpass dapat dipakai sebagai sarana untuk mematikan server dari jarak jauh secara terprogram.
- Perangkat lunak Ether-wake dapat bekerja secara baik untuk menghidupkan server dari jarak jauh dari keadaan mati.

## **B. Saran-Saran**

Salah satu kelemahan yang ada pada konfigurasi ini adalah apabila sumber listrik utama padam terlalu lama sampai batas waktu kemampuan kapasitas UPS memberi catu daya pada server pengelola. Intinya perlu dicari caranya agar server pengelola dapat bertahan menyala lebih lama. Saran-saran yang perlu disampaikan dalam penelitian ini adalah sebagai berikut.

- Perlu menggunakan UPS yang punya kapasitas lebih besar untuk mempertahankan server pengelola.
- Perlu dicari server pengelola yang hemat dalam menggunakan daya listrik, sehingga dapat bertahan menyala lebih lama untuk mengelola sistem.

### Daftar Pustaka

- [1] -----, 2003, *Ether-wake*, Linux man page
- [2] -----, 2008, *Sshpass, Linux man page*,  
<http://www.digipedia.pl/man/doc/view/sshpass.1>
- [3] Free Software Foundation, 2007, *Installation Instructions*, Dokumentasi Rups
- [4] Greenet, Inc. 2010, *Non-interactive ssh password auth*,  
<http://sshpass.sourceforge.net>
- [5] Mega System Technologies, Inc. 2003, *RUPS for Unix, UPS Monitoring Utility, User's Manual*, <http://www.megatec.com.tw>
- [6] Moh. Suryadiman, 2008, *Analisis Pengaruh UPS Terhadap Kinerja Perangkat Komputer*, BATAN
- [7] Wikimedia Foundation, Inc., 2010, *Wake on LAN*,  
[http://en.wikipedia.org/wiki/Wake on LAN](http://en.wikipedia.org/wiki/Wake_on_LAN)
- [8] Wikimedia Foundation, Inc., 2011, *Data Base Server*, [http://en.wikipedia.org/wiki/Database\\_server](http://en.wikipedia.org/wiki/Database_server)
- [9] Wikimedia Foundation, Inc., 2011, *File Server*,  
[http://en.wikipedia.org/wiki/File\\_server](http://en.wikipedia.org/wiki/File_server)
- [10] Wikimedia Foundation, Inc., 2011, *Name Server*,  
[http://en.wikipedia.org/wiki/Name\\_server](http://en.wikipedia.org/wiki/Name_server)
- [11] Wikimedia Foundation, Inc., 2011, *proxy Server*, [http://en.wikipedia.org/wiki/proxy\\_server](http://en.wikipedia.org/wiki/proxy_server)
- [12] Wikimedia Foundation, Inc., 2011, *Server*, <http://id.wikipedia.org/wiki/Server>
- [13] Wikimedia Foundation, Inc., 2011, *Uninterruptible Power Supply*,  
[http://id.wikipedia.org/wiki/Uninterruptible\\_power\\_supply](http://id.wikipedia.org/wiki/Uninterruptible_power_supply)
- [14] Wikimedia Foundation, Inc., 2011, *Web Server*, [http://en.wikipedia.org/wiki/Web\\_server](http://en.wikipedia.org/wiki/Web_server)