

PERANCANGAN TOOLS BERBASIS PYTHON UNTUK MEMANTAU KEAKTIFAN SERVER

Husni Thamrin

Jurusan Teknik Informatika, Fakultas Komunikasi dan Informatika

Universitas Muhammadiyah Surakarta

Email : husni@fki.ums.ac.id

ABSTRACT

Information system reliability can be measured by the availability of system when the information is needed. Server activeness is among the important factors to get reliable information system. Inactiveness of servers obstructs information availability and needs an early detection so that system administrator may quickly overcome the problem and restore the system.

A tools is needed to help a system admin detect server inactiveness, which alerts the admin of any possible problem. In this research, a tool is created using python programming language to periodically detect activeness of servers simultaneously. The result is logged into a file for later analysis. Whenever the tool detect inactiveness, it sends an email to the admin account and/or send a text message to a specified number.

Observation proves that the tools works as expected. The tools should be installed in one of the servers or in a computer close the server farm because network connectivity and computer location have a great influence to the monitoring activity.

Keywords : Server activity, monitoring, python

ABSTRAK

Kebandalan sistem informasi diukur dari ketersediaan informasi pada saat yang diperlukan dan keaktifan server merupakan faktor yang sangat penting untuk memperoleh sistem yang bandal. Ketidakaktifan server mengakibatkan pelayanan informasi terganggu sehingga harus dideteksi secara dini agar administrator dapat segera menanggulangi permasalahan dan pelayanan informasi segera dipulihkan.

Untuk mendeteksi ketidakaktifan server perlu dibuat tools yang dapat membantu admin memantau keaktifan server dan memberi tahu admin jika ada server tidak aktif. Pada penelitian ini dibuat tools berbasis bahasa python yang mendeteksi keaktifan sejumlah server secara simultan dan berkala. Hasil deteksi disimpan dalam berkas log dan jika dikenali adanya ketidakaktifan server, maka tools akan mengirim email dan atau pesan sms ke admin.

Hasil pengamatan menunjukkan bahwa tools dapat digunakan untuk mendeteksi keaktifan server. Tools sebaiknya diinstal pada komputer atau server yang paling dekat dengan server yang hendak diamati karena faktor konektivitas jaringan dan lokasi dapat mempengaruhi hasil pemantauan.

Kata kunci : keaktifan server, pemantauan, python

PENDAHULUAN

Kehandalan sistem informasi diukur sebagai perbandingan rentang waktu sistem secara aktual tersedia dibanding rentang waktu sistem diharapkan tersedia (Rauscher, 2000). Kehandalan sistem dapat diartikan sebagai tingkat ketersediaan informasi pada saat diungkin dan beroperasinya seluruh komponen yang menyusun sistem. Pada sistem informasi yang memanfaatkan server sebagai penyimpan dan pengelola informasi, operasionalitas server baik dari segi hardware maupun software menjadi parameter yang penting dalam menentukan kehandalan sistem informasi (Pickering, 2005). Server diharapkan dapat selalu berada dalam keadaan aktif dan perangkat lunak dan proses yang ada dalam server berjalan secara normal.

Ciri pertama dari beroperasinya sebuah server adalah bahwa server tersebut menyala (ON) dan dapat memberikan tanggapan jika dihubungi. Sebuah teknik standar yang digunakan untuk melihat beroperasinya sebuah server dari komputer lain adalah dengan memberikan perintah ping menuju nomor IP dari server yang diuji. Server yang menyala dan jalan (*up and running*) akan memberikan tanggapan jika mendapat perintah tersebut (Mansfield, 2003).

Tentu saja keadaan nyala dari sebuah server tidak harus berarti bahwa server tersebut mampu melayani dan memberikan informasi kepada pihak yang membutuhkan. Server web misalnya baru dapat memberikan informasi jika aplikasi *web server* seperti Apache atau IIS berjalan normal, berkas-berkas html dapat dibaca dengan baik dan *script-script* server dapat dieksekusi dengan benar (Pickering, 2005). Server basis data dapat memberikan pelayanan informasi dan perekaman data jika *database engine* berjalan

normal dan berkas-berkas penyimpanan data tidak ada yang rusak.

Oleh karena itu aktivitas pemantauan aktivitas server semestinya tidak hanya memberikan perintah ping ke IP dari setiap server namun juga menguji aktivitas proses yang ada dalam server. Hasil dari pengujian kemudian disampaikan kepada operator atau admin yang bertugas memantau kinerja server agar kesalahan dapat segera dianalisa dan perbaikan dapat segera dijadwalkan (Reilly, 2000).

Tulisan ini mendiskusikan perencanaan piranti lunak menggunakan python untuk memantau keaktifan server dalam suatu jaringan dan mendapatkan informasi yang menjadi indikator berjalannya proses di dalam server. Deteksi keaktifan server mengandalkan proses ping dan deteksi keaktifan *web server* menggunakan perintah akses halaman web. Hasil pengamatan dan analisa menunjukkan bahwa keaktifan server dapat dideteksi dengan program yang dirancang namun faktor konektivitas jaringan dan komputer pemantau sangat mempengaruhi hasil pengamatan.

METODOLOGI PENELITIAN

Penelitian dimulai dengan perancangan program pemantau keaktifan server untuk dijalankan secara berkala dan terus menerus. Program tersebut dibuat dengan bahasa pemrograman python dan dijalankan di komputer pemantau. Posisi komputer dan server dalam jaringan tampak pada Gambar 1. Server dan komputer pemantau terletak pada gedung yang berbeda yang berjarak sekitar 100 meter. Kedua gedung dihubungkan dengan kabel optik sedangkan jaringan komputer di dalam gedung menggunakan kabel ethernet.

Program pemantau dijalankan di komputer yang menjalankan sistem operasi berbasis UNIX. Program dijalankan secara berkala yaitu tiap 5 menit dan keberkalaan diatur dengan memanfaatkan fasilitas sistem operasi yaitu proses *cron*. Gambar 2 memperlihatkan diagram aktivitas program pemantau mulai dari inisiasi tugas *cron* sampai dilakukan *logging* data. Pada pengamatan kali ini, program dijalankan selama satu minggu penuh, dari hari Selasa hingga Selasa berikutnya.

Secara umum aktif atau tidaknya sebuah server ditandai dengan adanya tanggapan terhadap perintah ping. Keaktifan server web diuji dengan mengakses sebuah halaman web. Idealnya, halaman yang diakses dibuat secara khusus sebagai halaman web uji (*web test page*) agar data yang ditransfer dalam pengujian cukup kecil sehingga dapat dikirim dengan satu paket IP. Sebagai catatan, halaman muka sebuah situs dapat berukuran puluhan kilobytes belum termasuk gambar atau berkas pendukung lainnya. Implementasi akses halaman uji dengan python cukup sederhana sebagaimana tampak pada Gambar 3.

Sekali perintah ping terhadap sebuah server maupun pembacaan halaman web uji dapat memakan waktu beberapa milidetik hingga beberapa detik. Lama waktu pengujian tergantung pada lokasi server dalam jaringan dan apakah server yang diuji sedang aktif atau tidak. Pengujian server yang tidak aktif membutuhkan waktu yang jauh lebih panjang karena program pemantau perlu menunggu beberapa saat sebelum menyimpulkan bahwa server tidak memberi tanggapan. Jika terdapat banyak server yang harus dipantau dan dideteksi, proses pemantauan keseluruhan server akan memakan waktu yang lama jika dilakukan satu per satu (secara

batch). Untuk menghindari masalah waktu ini, digunakan teknik *threading* dan setiap proses pengujian satu server dilakukan oleh satu *thread*. Penggunaan teknik *threading* dapat menghemat waktu pengerjaan program pemantau secara signifikan (Matloff, t.t.). Program python yang digunakan merupakan modifikasi dari program yang dibuat oleh Well House Consultant (t.t.) di mana *thread* diimplementasikan sebagai objek dengan *property* *ip* untuk mencatat alamat IP dan *property* *status* untuk mencatat hasil ping server (lihat Gambar 4).

Hasil deteksi keaktifan server oleh setiap *thread* dikumpulkan dalam sebuah *list*. Setelah seluruh *thread* dijalankan, program pemantau merekam hasil deteksi ke sebuah berkas log. Penulisan ke berkas log dilakukan dengan memanfaatkan *library* yang telah ada yaitu *logging*.

Jika terdapat server yang tidak memberi tanggapan, selain dicatat dalam berkas log, dilakukan juga pengiriman email ke akun administrator dan atau pengiriman sms (*short message service*) ke nomor *handphone* yang dipilih.

Pengiriman email dilakukan dengan memanfaatkan *library* *smtplib*. Library ini memiliki berbagai objek untuk pengiriman email menggunakan protokol-protokol standar seperti SMTP (*simple mail transfer protocol*). Objek *smtp* yang digunakan untuk mengirim pesan email dibungkus dalam format MIMEText.

Pengiriman sms dari server ke nomor *handphone* dilakukan dengan menggunakan perangkat *broadband modem*. Tentu saja modem harus dilengkapi dengan kartu SIM yang masih aktif. Toolkit yang digunakan dalam proses pengiriman sms adalah *gammu*.

Gammu sebenarnya merupakan paket aplikasi yang berdiri sendiri dan dapat dijalankan melalui terminal sistem berbasis UNIX. Dalam penerapan yang dilakukan oleh penulis, gammu dikendalikan melalui program python. Hal ini dimungkinkan berkat adanya teknologi *ping* pada sistem operasi GNU/Linux dan dukungan program python untuk menerapkan teknologi tersebut. Contoh sepotong program python untuk mengirim sms melalui toolkit gammu dapat dilihat pada Gambar 5.

Sebelum digunakan untuk komunikasi, gammu harus disetting agar konfigurasi yang digunakan sesuai dengan kondisi yang ada. Bentuk konfigurasi itu antara lain adalah penentuan port di mana *broadband modem* ditancapkan dan kecepatan transfer data antara komputer dan modem. Setting konfigurasi pada gammu dapat dilakukan melalui perintah "gammu-config" pada terminal. Perintah tersebut akan menjalankan program interaktif di mana pemakai dapat dengan mudah memilih parameter-parameter setting yang diinginkan.

Data yang terdapat dalam berkas log kemudian dapat dianalisa untuk mengkalkulasi tingkat keaktifan dari server-server yang dipantau. Untuk proses analisa, berkas log yang ada di server sebaiknya disalin ke komputer pribadi. Penulis dalam konteks ini tidak membuat program khusus untuk menganalisa data pada berkas log dan memilih menggunakan *spreadsheet* untuk membuat tabel maupun grafik untuk dianalisis.

HASIL PENGAMATAN

Pada pengamatan kali ini, program digunakan untuk memantau empat buah server, tiga server milik Universitas, yang

dalam tulisan ini diberi label server A, server B dan server C, dan satu server milik *provider* internet terkemuka di dunia. Masa pengamatan adalah 7 hari dan proses deteksi dilakukan setiap 5 menit. Jumlah data yang terkumpul adalah 2016 data untuk tiap server. Jika dikurangi data dalam rentang jaringan tidak aktif (22.00 – 04.00 WIB), jumlah data menjadi 1512 buah untuk tiap server.

Analisis berkas log memperlihatkan bahwa faktor koneksi jaringan antara komputer pemantau dengan server cukup mempengaruhi hasil pemantauan. Data memperlihatkan bahwa terdapat beberapa kali perintah ping yang tidak mendapat tanggapan dari seluruh server yang dipantau pada satu waktu yang sama. Kejadian ini terlihat baik pada waktu jaringan sibuk maupun tidak. Ketidadaan tanggapan ping dari seluruh server sulit dianggap sebagai ketidakaktifan server dan lebih tepat jika kejadian ini diasumsikan sebagai kegagalan koneksi jaringan. Dengan asumsi ini, kegagalan koneksi jaringan dapat dihitung yaitu sebanyak 22 kali atau sebesar 1.46%. Jika dicermati secara harian, prosentase kegagalan koneksi jaringan tampak pada Gambar 6. Prosentase kegagalan koneksi dapat diterjemahkan sebagai reduksi tingkat ketersediaan informasi yang dapat dilayani oleh server sehingga mempengaruhi tingkat kehandalan sistem informasi secara keseluruhan.

Jika kegagalan koneksi jaringan tidak diperhitungkan (dianggap tidak ada) dalam proses pemantauan server, data menunjukkan bahwa seluruh server yang ada di Universitas dapat memberikan respon pada setiap perintah ping, artinya tingkat keaktifan server di Universitas adalah 100%. Respon tersebut terbagi menjadi dua kategori yaitu respon penuh dan respon parsial. Respon

parsial terjadi jika server menjawab satu dari dua perintah ping berurutan sedangkan respon penuh terjadi jika server menjawab kedua permintaan ping berurutan. Respon parsial umumnya muncul pada jam-jam sibuk. Gambar 7 memperlihatkan tingkat kemampuan server memberi respon penuh dibanding jumlah perintah ping. Kegagalan merespon penuh mengindikasikan beban kerja server atau jaringan. Pada hari ke-1, ke-2, ke-3 dan ke-7 (yaitu hari Selasa, Rabu, Kamis dan Senin) tingkat penggunaan sistem informasi sangat tinggi dan server cukup sibuk sehingga banyak terjadi kegagalan merespon penuh. Pada hari ke-5 dan ke-6, yaitu hari Sabtu dan Minggu, server dapat merespon secara penuh yang mengindikasikan rendahnya pemanfaatan sistem informasi pada kedua hari itu. Jika dirata-rata, kemampuan merespon penuh adalah 98,8%, di mana pada hari kerja angka ini turun menjadi 98,3% dan pada akhir minggu meningkat menjadi 100%.

Sekali lagi bahwa tingkat kehandalan sistem informasi diukur dari ketersediaan informasi dibanding masa aktif sistem. Faktor konektivitas maupun kesibukan server dapat menyebabkan informasi gagal diperoleh oleh pengguna. Ketidakmampuan server memberi respon secara penuh dan konektivitas jaringan dapat menyebabkan informasi terhambat atau gagal diterima oleh pengguna. Kedua faktor ini dapat diterjemahkan sebagai ketidakstabilan koneksi ke server. Gambar 8 memperlihatkan tingkat ketidakstabilan koneksi ke berbagai server yang dipantau selama masa pengamatan.

ANALISIS DAN DISKUSI

Sebagaimana diungkapkan oleh Rauscher (2000), kehandalan sistem informasi

diukur dari rasio rentang waktu aktual sistem berjalan dibanding rentang waktu yang diharapkan sistem tersedia. Kehandalan sebesar 99 persen belum dianggap sebagai angka yang tinggi karena dengan angka tersebut berarti sistem tidak berjalan (*down*) selama sekitar delapan puluh tujuh jam dalam setahun. Penulis belum menemukan pernyataan tentang angka kehandalan yang dianggap tinggi, oleh karena itu diasumsikan bahwa angka 99,9 persen kehandalan dikategorikan sangat tinggi sedangkan angka di bawah 99 persen dikategorikan rendah.

Hasil pengamatan terhadap tiga buah server yang ada di Universitas Muhammadiyah Surakarta selama periode penelitian menunjukkan bahwa seluruh server mempunyai keaktifan sebesar 100 persen dan angka ini dikategorikan sangat tinggi.

Perlu diingat bahwa perhitungan angka kehandalan yang dilakukan dalam penelitian ini berdasarkan pada tanggapan terhadap perintah ping dari sebuah komputer ke server yang dipantau. Sedangkan tanggapan terhadap perintah ping dipengaruhi oleh kondisi di luar server, misalnya konektivitas jaringan dari komputer pemantau sampai server dan faktor eksternal yang mempengaruhi kinerja dari komputer pemantau (Pickering, 2005). Oleh karena itu, jika program pemantau tidak menerima tanggapan dari seluruh server yang dideteksi maka dapat disimpulkan bahwa koneksi jaringan sedang mengalami masalah. Data hasil pemantauan pada periode tersebut tidak digunakan untuk menghitung kehandalan server.

Penggunaan perintah ping untuk mendeteksi keaktifan server akan menghasilkan data dengan makna yang berbeda dari yang diinginkan. Jika komputer pemantau berada jauh dari server yang dipantau, faktor koneksi

sangat mempengaruhi hasil pemantauan. Salah satu server yang dipantau pada penelitian ini merupakan server milik perusahaan penyedia layanan internet terkemuka di dunia. Keandalan sistem yang dimiliki perusahaan tersebut diakui oleh banyak pihak di seluruh dunia. Namun ternyata angka keandalan yang diperoleh melalui penelitian ini hanya 90,8 persen dan angka ini lebih kecil dari angka keandalan server di Universitas. Dapat ditarik kesimpulan bahwa angka keandalan tersebut merupakan angka keandalan sistem informasi secara umum, bukan angka keandalan server. Jika diasumsikan server selalu berada dalam keadaan bekerja, maka angka tersebut dapat dinyatakan sebagai keandalan koneksi dari komputer pemantau ke server yang dituju.

Dengan melihat faktor kegagalan koneksi sebesar 1,46%, keandalan sistem informasi di Universitas yang diamati dari komputer pemantau adalah sebesar 98,54 %. Angka ini tergolong rendah. Tentu saja kesimpulan ini masih terlalu dini dan belum representatif. Untuk mengamati keandalan sistem informasi secara lebih luas, perlu dilakukan pengamatan dari berbagai komputer di berbagai gedung yang ada bahkan perlu diamati juga dari komputer di luar Universitas.

Untuk keperluan mengamati keaktifan server, program pemantau sebaiknya diletakkan pada salah satu server yang aktif (misalnya *mail server*) dan berada dekat dengan server-server lainnya (misalnya berada pada satu hub) agar hasil pengamatan tidak banyak dipengaruhi oleh faktor konektivitas.

Program pemantau yang digunakan dalam penelitian ini mempunyai beberapa kekurangan. Jika komputer pemantau karena suatu peristiwa menjadi tidak aktif atau

mengalami *crash*, maka program pemantau tidak dapat dijalankan. Administrator tidak dapat segera mengetahui adanya kegagalan sistem karena tidak ada *tools* yang secara otomatis memberi tahu admin tentang kegagalan tersebut. Terdapat dua alternatif solusi yang dapat diimplementasikan untuk mengatasi kekurangan ini. Solusi pertama adalah dengan menambahkan kode program yang secara rutin (misalnya sehari sekali) memberi tahu admin bahwa program pemantau bekerja normal.

Solusi kedua adalah dengan menjadikan salah satu mesin komputer sebagai pemantau cadangan. Jika program pemantau pada komputer utama sedang tidak bekerja, program pemantau cadangan secara otomatis aktif dan mengambil alih pekerjaan yang dilakukan program pemantau pada komputer utama. Solusi kedua ini membutuhkan adanya komunikasi antara program pemantau yang ada di komputer utama dan komputer cadangan. Untunglah python dilengkapi library yang dapat digunakan untuk mengimplementasikan konsep komunikasi ini.

Faktor konektivitas mempengaruhi proses pengiriman pesan tentang kegagalan server kepada administrator. Email tidak akan terkirim ketika koneksi tidak tersedia. Untuk itulah alternatif pengiriman pesan melalui sms menjadi penting.

KESIMPULAN

Program pemantau keaktifan server yang diimplementasikan dengan bahasa python seperti yang digunakan dalam penelitian ini dapat bekerja dengan baik dalam memantau keaktifan server. Namun konsep dan implementasi program masih memerlukan perbaikan. Program sebaiknya diletakkan di

salah satu server atau dekat dengan server yang dipantau. Direkomendasikan pula adanya program partner yang menjadi cadangan jika program utama tidak bekerja karena suatu hal, misalnya karena komputer tempat

program utama mengalami *crash*. Hasil pengamatan mengandung informasi tentang konektivitas jaringan sehingga analisis dan pengambilan kesimpulan tentang keaktifan server perlu dilakukan hati-hati.

DAFTAR PUSTAKA

- Mansfield, Niall, 2003. *Practical TCP/IP, Designing, Using, Troubleshooting, TCP/IP Networks on Linux and Windows*, London, Addison-Wesley.
- Norm Matloff, t.t. *Programming on Parallel Machines*, Davis, University of California. E-book non publikasi, diakses 30 Maret 2011 di <<http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>>
- Pickering, R., 2005. *Web Server Reliability, a White Paper*, Milton Keynes, ipcortex.
- Rauscher, Richard R., 2000. "Evaluating the Reliability of Information Systems in Healthcare" in *Proc. of 2000 AMLA Annual Symposium*, (ISBN: 1-56053-480-X).
- Reilly, Douglas J., 2000. *Inside Server-based Applications*, Washington, Redmond.
- Well House Consultants, t.t. *Python Threads - a First Example*, Wiltshire, UK. diakses 30 Maret 2011 di <<http://www.wellho.net/solutions/python-python-threads-a-first-example.html>>

BHAKTI - DHARMA - WASPADA