

RANCANG BANGUN PROTOKOL PADA LAPISAN APLIKASI UNTUK PEMANTAUAN KUALITAS UDARA MELALUI JARINGAN TCP/IP

Alex Surapati

Program Studi Teknik Elektro FT, Universitas Bengkulu

Alexsurapati1@yahoo.com

ABSTRAK

Untuk pemantauan kualitas udara dengan menggunakan komputer diperlukan suatu protokol (aturan) khusus, sementara itu protokol tersebut sampai saat ini belum tersedia. Tujuan dari penelitian ini adalah mendesain suatu protokol yang berguna untuk pemantauan kualitas udara dengan menggunakan komputer pada jaringan TCP/IP. Tahapan penelitian yang dilakukan meliputi penyusunan sebuah skema yang akan dipakai sebagai dasar pembuatan protokol, pengumpulan contoh data dari hasil pemantauan kualitas udara dan implementasi dengan bahasa pemrograman Java sedangkan untuk basis data menggunakan MySQL. Hasil penelitian menunjukkan bahwa dengan menggunakan protokol aplikasi khusus untuk pemantauan kualitas udara, pengiriman data dari stasiun pemantau (klien) ke stasiun pusat (server) dapat berjalan lancar. Dengan hasil pengiriman data pemantauan kualitas udara yang baik maka pengolahan data tersebut untuk tujuan tertentu dapat lebih cepat dan terpercaya.

Kata kunci : TCP/IP, MySQL, kualitas udara

ABSTRACT

A computer based air quality surveillance requires a special protocol, meanwhile the special protocol was limitedly available. The purpose of this study was to design a computer based air quality surveillance protocol in TCP/IP network. The research activities consisted of preparing a scheme as a frame work in protocol design, collecting data samples from air quality and implementing the scheme with Java and MySQL. The results show that the data of air quality was effectively transmitted from surveillance center to server center and finally data processing for other purposes proceeded rapidly.

Key words: TCP/IP, MySQL, air quality

PENDAHULUAN

Perkembangan teknologi dan industri dewasa ini ternyata juga membawa dampak negatif bagi kehidupan manusia, yaitu dengan adanya berbagai macam limbah yang dibuang ke lingkungan. Sektor transportasi yang merupakan faktor pendukung mobilitas penduduk dalam melakukan kegiatan juga ikut menyumbangkan bahan pencemar ke udara.

Masalah pencemaran udara ini merupakan masalah yang perlu segera ditangani secara berkesinambungan. Salah satu bagian dari pengendalian pencemaran udara tersebut adalah pemantauan terhadap kualitas udara. Pemantauan sebaiknya dilakukan oleh stasiun pemantau kualitas udara yang dapat beroperasi secara terus menerus dan

datanya dapat dipantau secara langsung untuk memberikan kemudahan dalam pengendalian. Dengan demikian tindakan preventif dan antisipatif bila emisi bahan pencemar udara melebihi ambang batas dapat segera dilakukan (Sembiring *et al.*, 2000).

Mekanisme pemantauan dirancang untuk dapat beroperasi secara otomatis serta terus menerus. Data yang diperoleh dari pengukuran dikirim ke stasiun pusat yang mengumpulkan semua data yang ada serta mengkoordinasikan semua stasiun pemantau. Metode yang diusulkan oleh Orr dan Miller (1998) untuk keperluan pemantauan lingkungan dapat dipergunakan teknologi komputer, yaitu dengan menggunakan komputer di stasiun pusat dan stasiun pemantau serta komputer publik. Mekanisme komunikasi yang dipakai pada pemantauan ini di

rancang dengan memanfaatkan teknologi internet.

Aturan komunikasi antar komputer diwujudkan dalam bentuk protokol. (Arpaia *et al.*, 1997). Pada pemantauan kualitas udara yang diteliti ini, diperlukan protokol yang mampu menangani pengiriman data hasil pemantauan dari stasiun pemantau ke stasiun pusat untuk dikelola (Baxter, 2000). Menurut Rose (2001) protokol aplikasi yang tersedia dipandang tidak dapat mengakomodasi keperluan tersebut, sehingga perlu dirancang sebuah protokol pada lapisan aplikasi yang sesuai untuk kebutuhan pemantauan kualitas udara, yaitu protokol komunikasi antara *server* dan klien stasiun pemantau, yang selanjutnya akan disebut *Simple Remote Measurement Protocol (SRMP)*.

Postel (1982) mengatakan, untuk komunikasi antara *server* dan klien pengguna digunakan *Hypertext Transfer Protocol (HTTP)*, sehingga pengguna dapat memakai *browser* yang umum digunakan untuk menampilkan data yang diperoleh dari *server* stasiun pusat.

Untuk perancangan basis data digunakan *MySQL*, karena *MySQL* memiliki kelebihan-kelebihan (DuBois, 2000) diantaranya: konektivitas tinggi, dapat diperoleh secara gratis, memiliki kecepatan tinggi dan relative mudah digunakan. Untuk pembuatan halaman *web* dinamik digunakan *PHP*, yaitu pengolah skrip yang dapat menghasilkan halaman informasi dengan dan mempunyai kemampuan mengolah permintaan dari klien serta berhubungan dengan *server* basis data untuk memperoleh data yang bersifat dinamik (Bakken, 2001).

Tujuan dari penelitian ini adalah mempelajari, memahami dan mengembangkan suatu rancangan protokol komunikasi pada lapisan aplikasi di atas protokol *TCP/IP* untuk pemantauan kualitas udara melalui internet.

METODE PENELITIAN

Materi yang digunakan dalam penelitian ini antara lain bahan acuan tentang penelitian dan pelaksanaan pemantauan kualitas udara, baik perangkat keras maupun perangkat

lunak dan bahan acuan tentang penelitian bidang desain protokol lapisan aplikasi serta bahasa pemrograman Java, MySQL, PHP, dan JGraph. Alat yang digunakan berupa dua buah komputer, *modem* dan *network card*.

Penelitian ini dilaksanakan dengan melakukan perancangan protokol aplikasi antara *server* stasiun pusat dan klien stasiun pemantau. Implementasi protokol dapat dibuat dengan cara pembuatan prototip program, baik di sisi *server* maupun di sisi klien, serta memanfaatkan *HTTP server* di stasiun pusat untuk menangani permintaan informasi dari klien pengguna.

Hambatan yang timbul saat melakukan penelitian ini antara lain, tidak lengkapnya literatur atau buku-buku pendukung yang berhubungan dengan penelitian ini. Kegagalan dalam pengujian program pada jaringan internet juga menjadi hambatan dalam penelitian.

HASIL DAN PEMBAHASAN

Program *server* pada stasiun pusat menangani protokol *HTTP* dan *SRMP*, serta menangani transaksi data ke sebuah *server* basis data *MySQL*. Mengingat komunikasi dengan klien pengguna melalui protokol *HTTP* sudah ditangani oleh program *Apache*, maka program di *server* yang dibuat adalah program untuk mengimplementasikan protokol *SRMP*.

Pada prototip yang dapat dibuat sebagai bentuk implementasi, terdapat sebuah kelas utama yang diberi nama **ServerSRMP**, yang mula-mula melakukan inisialisasi untuk mengatur konfigurasi *server* dan memeriksa apakah *server* basis data siap dioperasikan. Pengaturan konfigurasi dilakukan dengan membaca *file ServerSRMP.cfg* baris demi baris dan memasukkannya ke dalam variabel yang sesuai, seperti terlihat dalam potongan berikut :

```
BufferedReader FileCFG =
    New BufferedReader (new FileReader
        ("ServerSRMP.cfg"));
String Item;
While ((Item = FileCf.readLine ()) != null {
    If (Item.equals ("PortSRMP"))
```

```

PortSRMP = Integer.parseInt
(FileCfg.readLine ());
else if (Item.equals (“[NamaServer]”))
    NamaServer = FileCfg.readLine ();
else if (Item.equals (“[ServerMySQL]”))
    ServerMySQL = FileCfg.readLine ();
else if (Item.equals (“[BasisData]”))
    BasisData = FileCfg.readLine ();
else if (Item.equals (“[Pengguna]”))
    Pengguna = FileCfg.readLine ();
else if (Item.equals (“[KataKunci]”))
    KataKunci = FileCfg.readLine ();
)
FileCfg.close ();

```

Pemilihan nomor *port* untuk protokol *SRMP* dapat dilakukan dengan pertimbangan bahwa nomor *port* tersebut belum digunakan untuk layanan standar, yang penting nomor *port* ini harus disepakati dengan klien stasiun pemantau sehingga komunikasi dapat berjalan dengan baik. Apabila ada klien yang menghubungi *server*, maka program membuat *socket* baru untuk berkomunikasi dengan klien tadi, kemudian membuat *instance* kelas *KoneksiSRMP* dengan *socket* tersebut sebagai parameter dan menjalankannya. Selanjutnya *server* kembali menunggu perintah koneksi berikutnya.

Supaya dapat menangani banyak permintaan dari beberapa klien secara simultan, maka kelas *KoneksiSRMP* merupakan tu-

runan dari kelas *Thread*. Kelas ini menggunakan *socket* yang dibuat oleh *server* tadi untuk merespon klien. Sesuai dengan protokol yang telah dirancang, kelas ini memanfaatkan informasi status untuk mengetahui posisi komunikasi yang sedang terjadi dalam alur protokol yang sedang berjalan. Oleh karena itu, pada kelas ini didefinisikan status yang akan dipakai sepanjang terjadinya komunikasi antara *server* dengan klien *SRMP*. Cara menyimpan data ke dalam basis data dilakukan dengan membuka koneksi ke *server MySQL* dan mengirimkan perintah *SQL* di dalam *method* *SimpanDataPengukuran*.

Pada program klien di stasiun pemantau hanya terdapat sebuah kelas, yaitu *Pemantau* yang menangani akuisisi data dari sensor serta pengiriman data ke *server* menggunakan protokol *SRMP*. Secara umum, program utama dari kelas ini adalah membaca *file* konfigurasi dan kemudian memeriksa waktu pada saat itu. Jika waktu pengukuran sudah terlampaui, maka dilakukan akuisisi data dari sensor menggunakan *method* *Akuisisi* dan kemudian data yang diperoleh dikirim ke *server SRMP* menggunakan *method* *KirimData*.

Berikut ini adalah sebuah contoh saat terjadi komunikasi pengiriman data hasil pengukuran kadar NO_2 , O_3 , CO dari klien pemantau (K) ke stasiun pusat atau *server* (S) :

```

S : +OK Selamat datang di server SRMP Stasiun Pusat.
K : ID Bengkulu.
S : +OK Bengkulu, silahkan masukkan password anda.
K : PASS tercemar.
S : -ERR Password salah, ulangi masukkan ID atau QUIT

K : ID Bengkulu.
S : +OK, silahkan masukkan password anda.

K : PASS terkenal.
S : +OK Password diterima, anda masuk ke status transaksi.

K : DATA 2004-3-19 10:15:00 NO2 120.

```

S : +OK Data diterima.
 K : DATA 2004-3-19 10:15:00 O3 O₃ 27.
 S : +OK Data diterima.

K : DATA 2002-3-19 10:15:00 CO 6.
 S : +OK Data diterima.
 K : CMND.

S : MSRT NO₂ 5.
 K : +OK Laju pengukuran konsentrasi NO₂ diatur setiap 5 menit.
 S : SDRT 60.
 K : +OK Laju pengiriman data diatur setiap 60 menit.

S : ENDC.
 K : QUIT.
 S : +OK Koneksi ditutup. Selamat tinggal.

Informasi paling utama yang disediakan untuk dipublikasikan kepada masyarakat harus memuat waktu pelaporan, ketentuan waktu, lokasi, Indeks Standar Pencemar Udara (ISPU) dari setiap parameter yang diukur.

KESIMPULAN

Berdasarkan hasil penelitian dan analisis diperoleh rancangan protokol pada lapisan aplikasi untuk komunikasi antara *server* stasiun pusat dan klien stasiun pemantau kualitas udara yang disebut dengan *Simple Remote Measurement Protocol (SRMP)*.

Perancangan protokol ini dapat diimplementasikan dengan membuat prototip program *server* stasiun pusat dan stasiun pemantau dan juga halaman *web* pemantauan kualitas udara yang ditampilkan pada *browser* di klien pengguna.

DAFTAR PUSTAKA

Arpaia, P., Baccigalupi, A., Cennamo, F., and Daponte, P. 1997. A Remote Measurement Laboratory for Educational Experiments. *Measurement*, 21(4): 157-169.

Bakken, S.S., Aulbach, A., Schmid, E., Winstead, J., Wilson, L.T., Lerdorf, R., Suraski, Z., Zmievski, A., and Atho, J., 2001. PHP Manual. PHP Documentation Group.

Baxter, Jr., J.F., 2000. Early Warning Detection and Notification Network for Environmental Condition. United States Patent Number 6,023,223.

DuBois, P., 2000. MySQL. New Riders Publishing, Indianapolis.

Orr, W.W., and Milloer, R.M.P. 1998. Method for Monitoring the Environment. United States Patent Numer 5,818,916.

Postel, J.B., 1982. Simple Mail Transfer Protocol. RFC 821, InterNIC Directory and Database Services.

Rose, M.T. 2001. On the design of Application Protocols. RFC 3117, InterNIC Directory and Database Services.

Sembiring, A.D., Ucu, and Purbo, O.W. 2000. Sistem Pemantauan Kualitas Udara Otomatis dengan Teknologi Paket Radio. <http://www.detik.com/net/onno/jurnal/004/aplikasi/ap-41.shtml>.