Mengirim e-mail dengan Java

Sasmito Adibowo

e-mail mungkin adalah aplikasi terpenting untuk Internet. Memang, pengguna e-mail lebih banyak dari para surfer web -- terutama pada saat krismon seperti ini :-). Memang, JavaSoft merencanakan API *Mail for Java*, tapi API tersebut tidak akan menjadi *core package* yang dijamin akan ada di setiap implementasi Java. Akibatnya kita harus membuat sendiri rutin e-mail. Untunglah, hal itu tidak terlalu sulit dilakukan.

Simple Mail Transport Protocol (SMTP)

Internic <http://www.internic.net/rfc> mendefinisikan SMTP dalam dua dokumen *Request for Comments* (RFC). RFC-821 mendefinisikan cara client dan server berkomunikasi dalam mengirimkan e-mail. RFC-822 mendefinisikan pemformatan e-mail dalam bentuk teks (sebelum ada MIME).

Di dalam SMTP, client menghubungi server lewat port 25. Si client mengirimkan perintah-perintah (seperti pada command prompt), dan server menjawab dengan kode status pada awal pesannya. Contoh pada paragraf berikut. Oh ya, server reformasi.or.id adalah -- sampai saat ini -- fiktif:

```
client --> (baris kosong)
server --> 220 mail.reformasi.or.id ESMTP Sendmail 8.8.7/8.8.7; Thu, 18 Jun 1998 13:39:59 +0700
client --> HELO localhost.localdomain
server --> 250 mail.reformasi.or.id Hello localhost.localdomain [198.168.1.100], pleased to meet
you
client --> MAIL FROM:  "Sasmito Adibowo" <adib@bitsmart.com>
server --> 250 adib@bitsmart.com... Sender ok
client --> RCPT TO: "id-reformasi" <milis-admin@reformasi.or.id>
server --> 250 milis-admin@reformasi.or.id... Recipient ok
client --> DATA
server --> 354 Enter mail, end with "." on a line by itself
client --> Date: Thu, 18 Jun 1998 08:13:21 +0700
client --> From: "Sasmito Adibowo" <adib@bitsmart.com>
client --> To: "Administrator Milis Reformasi" <milis-admin@reformasi.or.id>
client --> (baris kosong, untuk memisahkan header dengan body)
client --> Tolong kirimkan rencana kegiatan mendatang dan daftar website pro-client --> reformasi.
client --> Thanks berat.
client -->
client --> May The Force be with you,
client --> Sasmito Adibowo
client --> http://www.bitsmart.com/adib
client --> .
server --> 250 NAA00453 Message accepted for delivery
client --> QUIT
server --> 221 mail.reformasi.or.id closing connection
```

Karena platform Macintosh, DOS dan UNIX masing-masing mendefinisikan line-terminator yang berbeda ("\r" untuk Mac, "\r\n" untuk DOS+Windows+OS/2, dan "\n" untuk UNIX), RFC-821 memutuskan bahwa setiap perintah dan respons SMTP diakhiri oleh "\r\n" -- carriage-return *dan* linefeed. Karena itu, kita tidak bisa memakai fungsi PrintStream.println() pada Java, karena fungsi tersebut menggunakan akhir-baris yang berbeda pada setiap platform yang menjalankannya. Pemecahannya kita gunakan fungsi DataOutputStream.writeBytes().

JavaMail

Transaksi SMTP dibuat dalam suatu langkah-langkah transaksi, yang disimpan di dalam array String steps[][], yang akan dijalankan bila fungsi send() dipanggil. Untuk detailnya, silahkan lihat source JavaMail.java. di situ sudah saya beri comment yang (semoga) cukup memadai.

```
/**
 * JavaMail
 *
 * Mengirim e-mail dengan Java
 *
 * Copyright(C) Sasmito Adibowo <adib@bitsmart.com> 1997, 1998.
 *
 * CATATAN: Program ini _tidak_berhubungan_ dengan JavaMail API yang
 * sedang (sudah? kurang jelas kabarnya :-) ) dibuat oleh
 * Sun Microsystems.
 *
 */
```

```
import java.io.*;
import java.net.*;
import java.util.*;

public class JavaMail
{
   public  static final int SMTPPORT = 25;
   private static final int COMMAND = 0, REPLY = 1;

   /**
    * bila 'true', tulis jalannya transaksi SMTP ke debugStream
    */
   private boolean debugMode = false;

   /**
    * OutputStream bila debug == true
    * default ke System.err
    *
    * @see System.err
    *
    */
   private PrintStream debugStream = System.err;

   /**
    * menyalakan/mematikan debugging mode.
    *
    * @return mode sebelumnya
    *
    */
   public boolean setDebugMode(boolean mode) {
        boolean prevMode = debugMode;
        debugMode = mode;
        return prevMode;
   }

   /**
    * rubah debugging stream
    *
    * @return  stream debugging sebelumnya
    */
   public OutputStream setDebugStream(OutputStream stream) {

        OutputStream prevStream = debugStream;
        if(stream  instanceof PrintStream) {
            debugStream = (PrintStream) stream;
        }
        else {
            debugStream = new PrintStream(stream);
        }
        return prevStream;
   }

   /**
    * langkah-langkah transaksi SMTP
    *
    */
   private String[][] steps = null;

   /**
    * constructor untuk mengirimkan mail ke beberapa penerima
    *
    * @param from    nama user pengirim; harus dalam bentuk:
    *                "Nama user" <e-mail@host.domain>
    * @param toList  daftar nama penerima; format seperti pada parameter
    *                'from'
    * @param subject subyek e-mail, 7-bit ASCII, tidak boleh ada "\r\n"
    * @param body    isi e-mail. Tidak boleh ada '.' yang berdiri sendiri
    *                dalam satu baris.
    *
    * @throws UnknownHostException  bila gagal untuk mendapatkan nama
    *         local host (127.0.0.1)
    *
    * @see JavaMail(String,String,String, String);
    */
```

```
public JavaMail(String from,String[] toList, String subject, String body)
   throws UnknownHostException
{
     this.steps = genSteps(from,toList,subject,body);
}

/**
 * constructor untuk mengirimkan mail ke satu penerima
 *
 * @see JavaMail(String,String[],String, String);
 */
public JavaMail(String from,String to, String subject, String body)
   throws UnknownHostException
{
     String[] toList = new String[1];
     toList[0] = to;
     this.steps = genSteps(from,toList,subject,body);
}


/**
 * menghasilkan langkah-langkah untuk transaksi SMTP.
 *
 * @param from    nama user pengirim; harus dalam bentuk:
 *                "Nama user" <e-mail@host.domain>
 * @param toList  daftar nama penerima; format seperti pada parameter 'from'
 * @param subject subyek e-mail, 7-bit ASCII, tidak boleh ada "\r\n"
 * @param body    isi e-mail. Tidak boleh ada '.' yang berdiri sendiri
 *                dalam satu baris
 *
 * @return String[][] langkah-langkah transaksi SMTP. String[x][COMMAND]
 *         adalah perintah yang dikirimkan ke SMTP host; String[x][REPLY]
 *         adalah awal dari reply yang diperbolehkan (menandakan kondisi
 *         sukses);
 *
 * @see COMMAND
 * @see REPLY
 *
 * @throws UnknownHostException  bila gagal untuk mendapatkan nama
 *         local host (127.0.0.1)
 *
 * Error message SMTP didahului oleh suatu angka yang menyatakan kode
 * kesalahan, digit pertamanya sudah cukup untuk mengidentifikasi
 * kesalahan:
 * <ul>
 *    <li> 2 berarti sukses
 *    <li> 3 berarti diperlukan lebih banyak info
 *    <li> 4 dan 5 berarti gagal
 * </ul>
 *
 */
private String[][] genSteps(String from,String[] toList,
    String subject, String body) throws UnknownHostException {
    //
    // lamgkah-langkah awal untuk login  ke SMTP host
    //
    final String [][] loginSteps = {
         // knock, knock, anybody home?
         { "","2" },
         // permisi, nama saya ...
         { "HELO " + InetAddress.getLocalHost().getHostName() + "\r\n","2"},
         // pengen kirim pesan nih, dari ...
         { "MAIL FROM:" + from + "\r\n","2" },
    };

     //
     // langkah minimal ada 7 :
     //   null, HELO, MAIL FROM, RCPT TO, DATA, body, QUIT
     //
     //
     Vector commands = new Vector(7);
     Vector replies  = new Vector(7);

     // login ke remote host:..
```

```
        for(int i = 0; i < loginSteps.length; i++) {
            commands.addElement(loginSteps[i][COMMAND]);
            replies.addElement(loginSteps[i][REPLY]);
        }

        // kirimkan daftar pemgirim
        for(int i = 0; i < toList.length; i++) {
            commands.addElement("RCPT TO:" + toList[i] + "\r\n");
            replies.addElement("2");
        }

        // data
        commands.addElement("DATA\r\n");
        // kode pesan harus "354" untuk menandakan sukses
        replies.addElement("354");

        // kirim mail data
        commands.addElement(formatText(subject,body,true,true) + "\r\n.\r\n");
        replies.addElement("2");


        // logout
        commands.addElement("QUIT\r\n");
        replies.addElement("2");

        //
        // 'pack' langkah-langkah SMTP ke String[][]
        //
        String[][] steps= new String[commands.size()]
            [(COMMAND > REPLY ? COMMAND : REPLY) + 1];
        for(int i=0; i < commands.size(); i++) {
            steps[i][COMMAND] = (String) commands.elementAt(i);
            steps[i][REPLY]   = (String) replies.elementAt(i);
        }

        // selesai deh
        return steps;
    }

    /**
     * Lakukan pemformatan dasar
     *
     * @param body           teks yang akan diformat -- harus sudah ada mail header
     * @param includeDate    jadikan 'true' untuk memasukkan tanggal ke body
     * @param includeBanner  jadikan 'true' untuk memasukkan merk e-mailer.
     *
     */
    protected String formatText(String subject, String body,boolean includeDate,boolean
includeBanner)
    {

       StringBuffer data = new StringBuffer(body.length() + 512);

       //
       // tanggal di e-mail header
       //
       if(includeDate) {
         data.append("Date: ");
         data.append(new Date().toString());
         data.append("\r\n");
       }

       //
       // merk e-mail client
       //
       if(includeBanner) {
         data.append("X-Mailer: ");
         data.append(getClientBrand());
         data.append("\r\n");
       }

       //
       // subject
       //
```

```
      data.append("Subject: ");
      data.append(subject);
      data.append("\r\n");

      //
      // body
      //
      data.append(body);

      return data.toString();
   }

   public void send(String mailhost)
      throws MailException, UnknownHostException, IOException
   {
        send(mailhost,SMTPPORT);
   }

   /**
    * lakukan transaksi SMTP
    *
    * @param  mailHost  nama server SMTP
    * @param  port      nomor port; biasanya 25
    *
    * @throws IOException   bila host SMTP tidak dapat dihubungi
    *
    */
   public void send(String mailhost,int port)
      throws MailException, UnknownHostException, IOException
   {

      // hubungi server
      Socket socket = new Socket(InetAddress.getByName(mailhost),port);

      if(debugMode) debugStream.println("Server " + mailhost + " terhubungi");

      DataOutputStream out = new DataOutputStream(socket.getOutputStream());
      DataInputStream  in = new DataInputStream(socket.getInputStream());
      int state = 0;
      String error = null;

      // lakukan langkah-langkah
      while(state < steps.length)
      {
         // tulis ke SMTP host
         out.writeBytes(steps[state][COMMAND]);
         out.flush();

         if(debugMode) debugStream.println("kirim:\t" + steps[state][COMMAND]);

         // baca reply -- termasuk yang lebih baik dari satu baris
         String lookahead;
         do
         {
            lookahead = in.readLine();
            if(debugMode) debugStream.println("terima:\t" + lookahead);

         }
         while(lookahead.charAt(3) != ' ');

         // OK?
         if(lookahead.startsWith(steps[state][REPLY]))
            // lakukan langkah berikutnya
            state++;
         else
         {
            // terjadi kesalahan
            error = lookahead;
            // ketika hampir selesai, tetapi belum sampai QUIT
            state = state < steps.length - 1 ?
                    steps.length - 1:
                    steps.length;
         }
      }
```

```
   // laporkan kesalahan, bila ada
   if(null != error)
      throw new MailException(error);
}

/**
 * Merk e-mailer
 *
 */
protected String getClientBrand() {
     // format yang biasa dipakai oleh client e-mail adalah:
     // nama-program angka-revisi [bahasa] (platform; versi-negara)
     return "Arcle Technologies JavaMail 0.5 [id] (Java 1.1; ID)";

}

/**
 * fungsi main() bila dijalankan sebagai aplikasi. Fungsi ini juga
 * memberi contoh cara memakai JavaMail
 *
 */
public static void main(String[] args)
   throws Exception
{
 // mode flags
 boolean debug = false, quiet = false;

 // streams -- biar gampang kalo mau ngerubah
 PrintStream out   = System.out;
 PrintStream err   = System.err;
 BufferedReader in  = new BufferedReader(new InputStreamReader(System.in));

 // message settings
 String   from,subject,message,mailServer;
 String[] toList;

 // banner
 String banner =
     "Arcle Technologies JavaMail version 0.5\n" +
     "Copyright(C) Sasmito Adibowo <adib@bitsmart.com> 1997, 1998.\n";

 // parse command line
 for(int iter=0;iter < args.length; iter++) {
     if( args[iter].charAt(0) == '-' ){
         if(args[iter].equals("--quiet")){
             quiet = true;
         }
         else if(args[iter].equals("--debug")){
             debug = true;
         }
         else {
             out.println(banner);
             out.println(
                 "penggunaan: java JavaMail [-options]\n\n" +
                 "di mana options adalah:\n" +
                 "\t--debug \tcetak sesi SMTP ke standard error.\n" +
                 "\t--quiet \tjangan cetak prompt ke standard output.\n");
             System.exit(1);
         }
     }
 }

 if(!quiet) {
     out.println(banner);
 }

 //
 // tanya server
 //
 if(!quiet) out.println("Server SMTP:");
 mailServer = in.readLine();

 //
 // siapa yang ngirim?
```

```
//
if(!quiet) out.println("Alamat e-mail pengirim:");
from = in.readLine();


//
// daftar penerima
//
if(!quiet) out.println("Tulis daftar e-mail penerima.\n" +
    "Akhiri dengan '.' pada satu baris yang berdiri sendiri.");

Vector rcptList = new Vector(1,7);
for(;;){
    String to = in.readLine();
    if(to.equals(".")) {
        break;
    }
    else{
        rcptList.addElement(to);
    }
}
toList = new String[rcptList.size()];
rcptList.copyInto(toList);


//
// tentang apa?
//
if(!quiet) out.println("Subject e-mail:");
subject = in.readLine();


//
// tanya mail data
//
if(!quiet) out.println(
    "Masukkan header dan isi e-mail, dipisahkan oleh baris kosong.\n"+
    "\nAkhiri dengan '.' pada satu baris yang berdiri sendiri..");

StringBuffer msgBuf = new StringBuffer();
for(;;) {
    String buff = in.readLine();
    if(buff.equals(".")) {
        break;
    }
    else{
        msgBuf.append(buff);
        msgBuf.append("\r\n");
    }
}
if(!quiet) out.println("Data e-mail diterima.");
message = msgBuf.toString();


int portNum;
// parse port number from 'mailServer'
// format is  host:port
StringTokenizer st = new StringTokenizer(mailServer,":");
mailServer = st.nextToken();
try {
    portNum = Integer.parseInt (st.nextToken());
}
catch(Exception e) {
    // kalo gagal
    portNum = SMTPPORT;
}

//
// buat e-mail session
//
JavaMail mail = new JavaMail(from,toList,subject,message);

if(debug) {
    mail.setDebugMode(true);
```

```
    }

    //
    // hubungi server dan kirim pesan
    //
    try {
        if(debug) err.println("Menghubungi " + mailServer + ":" + portNum +
            " untuk mengirimkan surat...");
        mail.send(mailServer,portNum);
    }
    catch(MailException e) {
        err.println("Kesalahan SMTP pada server " +
            mailServer + ":" + portNum);
        err.println("pesan: " + e.getMessage());
        System.exit(3);
    }
    if(debug) err.println("Surat telah dikirimkan.");
    System.exit(0);
    }
}
```

```
import java.io.*;

public class MailException extends IOException
{
   public MailException(String message)
   {
      super(message);
   }
}
```

Fungsi format() dapat diperbaiki lagi, untuk memastikan tidak ada string "\r\n.\r\n" di dalamnya, yang akan mengakhiri pengiriman isi e-mail ke server dan pasti akan mengacaukan transaksi SMTP. Juga, fungsi ini tidak mengecek apakah *character set* e-mail adalah dalam 7-bit ASCII, yang merupakan keharusan dalam SMTP.

Selain itu, dapat juga dikembangkan dukungan untuk mengirim file biner melalui e-mail, via MIME atau UUENCODE. Plus user interface yang lebih memadai untuk end-user. Satu lagi, karena masalah security, untuk mengirim mail dari applet, server SMTP haruslah server yang hostname-nya sama dengan server web tempat asal applet tersebut. Karena applet tidak bisa membuat koneksi ke server lain selain server tempat ia berasal.

Salam,
Sasmito Adibowo
http://www.geocities.com/adibs