

LAYANAN INFORMASI AKADEMIK BERBASIS ELECTRONIC MAIL (E-MAIL)

A. Dicky Ferlanda, L2F099565
Jurusan Teknik Elektro, Fakultas Teknik
Universitas Diponegoro, Semarang

Abstrak-- E-mail merupakan sarana komunikasi bagi seluruh pengguna internet yang mudah dan efisien. Sehingga penggunaannya dapat diterapkan di segala bidang. Dalam bidang pendidikan e-mail dapat digunakan sebagai layanan sistem informasi akademik, lebih-lebih pada lembaga pendidikan yang telah mempunyai Sistem Informasi Akademik. Layanan e-mail yang terintegrasi dengan sistem informasi akademik dapat memberikan nilai lebih dan keuntungan tersendiri.

Dalam Tugas akhir ini dibangun suatu sistem mail autoresponder yang diterapkan pada Sistem Informasi Akademik. Sistem ini berfungsi untuk memberikan layanan informasi akademik melalui e-mail secara autoreply. Sistem autoresponder ini dibangun dengan dua program utama yaitu mail filter dan program PHP.

Layanan informasi akademik melalui e-mail ini ditujukan bagi mahasiswa, sehingga mampu memberikan kemudahan dalam mendapatkan informasi akademis. Mahasiswa yang ingin mendapatkan informasi akademik dapat langsung mengirimkan request-nya melalui e-mail. Dan akan segera mendapat informasi yang diinginkan tadi secara autoreply.

Keyword : autoresponse, autoreply, mail filter.

I. Pendahuluan

1.1 Latar Belakang

Dewasa ini penggunaan e-mail merupakan hal yang penting dalam perkembangan teknologi informasi. E-mail dari kata asalnya yaitu *electronic mail* mempunyai arti surat yang dikirimkan melalui media elektronik hal ini adalah komputer yang terkoneksi ke internet. Sehingga setiap pemakai internet dapat saling mengirimkan dan menerima surat melalui layanan internet yang satu ini.

Banyak sekali layanan internet berbasis web yang juga memberikan layanan e-mail. Sistem informasi akademik merupakan layanan internet berbasis web. Sistem akademik ini berfungsi untuk memberikan informasi mengenai semua yang berkaitan dengan lembaga pendidikan yang dimilikinya. Informasi ditujukan baik untuk pemakai internet secara umum maupun civitas akademik dalam lingkungan kampus.

Sistem informasi akademik pada universitas yang telah ada akan terasa lengkap jika dapat memberikan layanan informasi melalui berbagai cara, salah satunya melalui e-mail. Sistem informasi akademik yang telah memiliki mail server dapat diaplikasikan untuk memberikan informasi kepada mahasiswa melalui e-mail. Terutama bagi mahasiswa yang hanya mempunyai akses e-mail ketika koneksi ke internet, atau ketika mereka yang memiliki akses e-mail dengan handphonenya. Mahasiswa yang ingin

mendapatkan informasi tertentu dari SIA dapat mengirimkan e-mail dan menuliskan request yang dimaksud dalam baris *subject*. Kemudian mail server merespon request yang ditulis di *subject* e-mail dan mengirimkan reply informasi yang dibutuhkan tadi. Dalam layanan ini tidak perlu selalu terkoneksi dengan internet jika ingin mendapatkannya, cukup mengirim mail request dalam satu waktu, dan mengambil informasinya di lain waktu dengan hanya mengecek *mailbox*. Karena program akan membalas e-mail berisi informasi yang diminta ke alamat si pengirim tadi.

1.2 Tujuan

Tujuan dari Tugas Akhir ini adalah merancang sebuah program pada mail server dimana mail server yang telah ada pada web sistem informasi akademik dapat memberikan layanan informasi akademik secara *autoresponse* kepada mahasiswa. Request yang dikirim oleh mahasiswa dalam bentuk e-mail akan ditanggapi secara langsung dan memberikan *autoreply* informasi yang tadi di-request.

1.3 Pembatasan Masalah

Pada Tugas Akhir ini pembahasan akan dibatasi pada hal-hal berikut ini :

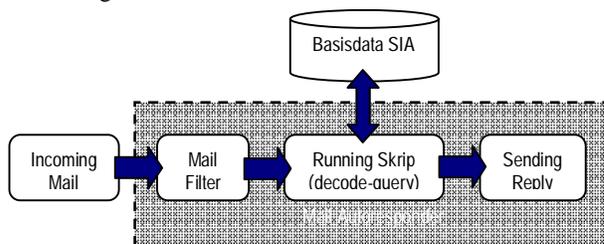
1. Dalam Tugas akhir ini dilakukan perancangan program guna memberikan layanan informasi akademik secara *autoreply* melalui e-mail.
2. Layanan informasi *autoresponse*-nya meliputi KRS, KHS, Jadwal Kuliah dan Jadwal Dosen, dan Transkrip Nilai.

3. Dalam tugas akhir ini basisdata yang digunakan adalah basis data sistem informasi akademik yang telah ada. Sehingga program hanya meng-*query* data, tidak melakukan perubahan atau pembuatan basis data baru.
4. Pada tugas akhir ini tidak membahas perancangan dan pembangunan mail server.

II. Layanan SIA Autoresponse via E-mail

Program layanan informasi akademik berbasis e-mail ini dibuat dan ditempatkan pada mail server. Program layanan ini merupakan cara alternatif bagi mahasiswa untuk mendapatkan informasi. Sehingga dapat mempermudah bagi mahasiswa untuk memperoleh informasi akademik yang mereka inginkan.

Ide dasarnya adalah program berfungsi untuk menangkap *request* dari mahasiswa berupa e-mail untuk diproses dan menghasilkan informasi yang akan dikirim kembali kepada mahasiswa yang mengirim e-mail tadi. Dalam mengambil data, program melakukan koneksi ke basisdata sistem informasi akademik yang telah ada. Apa yang dilakukan program adalah hanya melakukan *query* data dalam basisdata SIA untuk menyusun informasi. Program layanan ini dapat dimodelkan sebagai berikut:



Gambar 2.1 Pemodelan Sistem Mail Autoresponse

2.1 Mail Autoresponse

Mail autoresponse merupakan suatu sistem dimana dapat merespon secara otomatis e-mail yang masuk ke suatu alamat e-mail (*mail server*) dan kemudian melakukan *autoreply*. *Autoresponder* adalah kebalikan dari mengirim e-mail, disini terdapat program yang akan dijalankan ketika menerima sebuah e-mail. *Autoresponder* sudah banyak diaplikasikan dalam dunia internet, sebut saja yang mudah yaitu di yahoo. Ketika kita mendaftar menjadi user dalam yahoo mail kita secara langsung mendapat e-mail yang berisi sambutan dari yahoo mail itu sendiri. Ini merupakan salah satu penerapan *mail autoresponse*. Kunci dalam membuat *autoresponder* adalah memahami bagaimana e-mail diterima di server.

Dalam men-*delivery* e-mail yang masuk bisa digunakan suatu *mail processing program* yang

fungsinnya mem-filter e-mail untuk menjalankan suatu skrip, salah satunya yaitu *Procmail*. Program/skrip yang dieksekusi nantinya akan mendapatkan e-mail dari *stdin* dan berbagai informasi *headers* yang berada di *variable-variabel environment* tertentu.

Procmail adalah *mail processing program* yang ada di UNIX, dapat membantu menyelesaikan tugas-tugas berikut, antara lain:

1. Mem-filter e-mail yang masuk (misalnya memisahkan pesan mailing-list dari pesan-pesan yang lain), sangat berguna untuk *preprocessing* e-mail yang masuk dalam jumlah besar.
2. Mengurutkan e-mail yang masuk berdasarkan kriteria, seperti pengirim, ukuran pesan atau keyword dalam *subject:baris*.
3. Mengimplementasikan mekanisme *autoreply* seperti membalas e-mail kepada siapa yang mengirim pesan secara otomatis.

Procmail dapat bekerja dengan cara membaca file yang dibuat dalam home directory, file ini bernama **.procmailrc**. File ini berisi instruksi (*resep*) untuk pemrosesan e-mail yang masuk. Jika sebuah pesan cocok dengan kondisi yang ditulis, maka *procmail* mengintersep pengiriman normal berdasar spesifikasi yang ditulis tadi. Misalnya, pesan dapat diarahkan ke program, atau di-forward ke penerima yang lain, bahkan mungkin untuk menghasilkan *autoreply*. Jika tidak satupun kondisi yang cocok, maka *procmail* akan menaruh pesan ke *inbox default*.

Sebuah file *.promailrc* memiliki 2 bagian, yaitu *assignments* dan *recipes*. Apa saja yang didahului dengan tanda (#) adalah komentar, dan akan diabaikan. *Assignment* yang berada pada posisi atas berfungsi untuk men-setup variabel-variabel sehingga *procmail* tahu keberadaan program dan mailbox. *Recepies* merupakan pekerjaan filtering yang sesungguhnya berlangsung. *Recepies* mempunyai format sebagai berikut:

```

:0 [flags] [: [lock-file] ]
zero or more conditions
one action line
  
```

Idenya, bahwa jika syarat (*condition*) terpenuhi, aksi (*action*) akan dilaksanakan. Disini *procmail* dapat langsung digunakan untuk melakukan *mail autoreply* sederhana. Namun untuk sistem *mail autoresponse* yang melibatkan *query* data, dan penyusunan informasi perlu digunakan skrip program yang lain.

Suatu skrip dapat dijalankan dengan mengarahkan e-mail masuk yang telah di-filter, sehingga e-mail tadi menjadi trigger maupun input dari skrip yang telah dibuat. Disini *.procmailrc* melakukan penyeleksian berdasar *subject* pesan. *Subject* ditangkap untuk mengetahui request apa

yang diinginkan si pengirim. Selanjutnya e-mail langsung diarahkan ke masing-masing folder dan skrip berdasarkan subject tadi. Dalam tugas akhir ini skrip dibuat dengan menggunakan bahasa pemrograman PHP untuk memproses request berupa e-mail.

2.2 Skrip PHP

E-mail request yang sudah dialihkan procmail ke skrip program PHP ditangkap sebagai input (`stdin`). Dalam mengolah request berupa e-mail maka program perlu men-decode pesan agar diperoleh headers dan body dari e-mail. Terdapat suatu modul/paket dari PEAR (PHP Extension and Application Repository) yaitu `Mail_Mime` yang dapat berguna untuk men-decode e-mail sehingga dapat mempermudah proses identifikasi pesan.

Selanjutnya dalam membalas mail request diperlukan fungsi `mail()` yang telah ada dalam PHP. PHP otomatis dapat memilih mengirimkan e-mail menggunakan program eksternal seperti `/usr/sbin/sendmail` yang default di Linux dan UNIX. Sintaks `mail()` yang digunakan untuk mengirim sebuah e-mail adalah sebagai berikut:

```
mail($to, $subject, $message,
    [$additional_headers,
    [$additional_parameters]]);
```

`$additional_headers` adalah sebuah string yang terdiri dari satu atau lebih baris header tambahan yang ingin disebutkan. Biasanya yang paling sering disebutkan di sini adalah header `From` atau `Reply-To`. `$additional_parameters` berguna untuk mengirim opsi ke program `/usr/sbin/sendmail` atau yang setara.

2.3 PEAR

PEAR (PHP Extension and Application Repository) merupakan repositori resmi dari php.net yang salah satu tujuannya untuk menyediakan library dan paket bagi pengguna PHP dengan kode-kode yang *open source*. Kelas-kelas dasar PEAR dan PEAR manager (`pear`) umunya disertakan dalam distribusi standar PHP.

Dalam paket `Mail_Mime` terdapat beberapa kelas, salah satu kelas yang digunakan adalah `Mail_mimeDecode::decode()`. `Mail_mimeDecode` berisi fungsi-fungsi (*functions*) untuk membuat object `Mail_mimeDecode` baru, men-decode dan menghasilkan struktur yang berisi data pesan. E-mail yang diarahkan ke skrip menjadi input string bagi `Mail_mimeDecode` ini untuk kemudian di-decode, hasilnya berupa array misalnya `$structure`. Header-header disimpan di dalam `$structure['headers'][$<nama-header>]`, dimana `<nama-header>` adalah nama header yang telah dikonversi menjadi huruf kecil semua. Dengan

demikian dapat diperoleh header pengirim (From), dan isi pesan (Body) guna pengolahan request.

Header pengirim diambil untuk mendapatkan alamat e-mail si pengirim guna membalas e-mail (reply). Sedangkan body untuk keperluan verifikasi jika dibutuhkan, yaitu NIM dan Password. Dalam mengurai isi pesan dibutuhkan mekanisme regex (*regular expression*) sederhana. Hal ini berfungsi untuk pemisahan NIM dan Password sekaligus pencocokan formatnya, misalnya NIM harus memiliki format L2FXXXXXX (X = berupa angka tanpa spasi).

2.4 Query Basis

Query basisdata digunakan untuk mengumpulkan data dari suatu basisdata untuk menghasilkan informasi yang dibutuhkan. Sebelum melakukan query basisdata maka hubungan dengan server basisdata harus tercipta.

Tanpa melakukan koneksi dengan server basisdata, maka tidak bisa untuk mengambil data dari basisdata tersebut (*query* basisdata). Untuk melakukan koneksi dengan server basisdata (MySQL), PHP menggunakan fungsi `mysql_connect()`, sintaks yang digunakan adalah:

```
mysql_connect(hostname, username,
    password);
```

Koneksi ke basisdata akan secara otomatis terputus ketika skrip program selesai dijalankan seluruhnya, kecuali diberikan fungsi `mysql_close()`. Fungsi koneksi diatas akan memberikan nilai boolean TRUE apabila koneksi berhasil dan FALSE apabila gagal.

Apabila pengguna telah berhasil melakukan koneksi dengan server basisdata MySQL, maka pengguna dapat melakukan *query*. PHP tidak menyediakan fungsi khusus untuk melakukan operasi data, sehingga *query* yang digunakan adalah sintaks-sintaks dari MySQL. Tahapan untuk melakukan *query* data adalah sebagai berikut:

1. Koneksi ke basisdata
2. Memilih basisdata yang akan diambil datanya.
3. Mendeklarasikan sebuah variabel string yang berisi sintaks perintah (*query*) MySQL yang akan dilakukan.
4. Melaksanakan sintaks MySQL menggunakan fungsi `mysql_query()`, jika sintaks yang akan dijalankan menghasilkan output yang akan ditampilkan, maka harus dideklarasikan sebuah variabel untuk menampung hasil tersebut.
5. Mengambil hasil dari sintaks MySQL yang dilaksanakan menggunakan fungsi-fungsi seperti `mysql_fetch_array()`, `mysql_fetch_row()`, `mysql_num_rows()`, dan lain-lain tergantung pada format output yang diinginkan.

Sintaks yang digunakan untuk mengambil data dari basisdata tertentu adalah sebagai berikut:

```
SELECT field1,field2, ... FROM
nama_table WHERE condition1,
condition2, ... ORDER BY field
```

Parameter `ORDER BY` untuk mengurutkan data berdasar field. Standar urutannya adalah dari angka terkecil atau dari huruf A, dan dari data pertama masuk hingga data terakhir. Untuk mendapatkan urutan yang terbalik maka dibutuhkan atribut `DESC`.

III. Perancangan Program

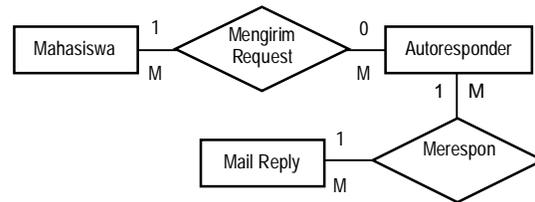
Perancangan program layanan informasi akademik berbasis e-mail ini dilakukan dengan *Entity-Relationship Diagram* (E-R Diagram), *Flowchart* dan *Algoritma Program*, dan *Data Flow Diagram* (DFD). Seperti yang dijelaskan pada bab 2, sistem *autoresponse* dalam program ini memiliki ide dasar bahwa program berfungsi untuk menangkap *request* dari mahasiswa berupa e-mail, mem-filter e-mail, men-decode e-mail, verifikasi, melakukan query basisdata, dan mengirimkan informasi.

Sistem layanan informasi akademik ini sebenarnya terdiri atas dua program yaitu *mail filtering* dan program *autoresponse*. Dalam mem-filter e-mail yang masuk akan digunakan procmail. Procmail bertujuan untuk mengarahkan e-mail ke program *autoresponse* dan mailbox tersendiri berdasarkan subjeknya. Sedangkan program *autoresponse* dibangun dengan menggunakan PHP mode biner. Program akan berjalan otomatis ketika mendapat input e-mail yang diarahkan kepadanya dari procmail. Kemudian program akan menanggapi e-mail tersebut dan mengirim balasan berupa informasi yang diinginkan. Misalnya terdapat e-mail masuk dengan subjek KRS, maka procmail akan menaruh e-mail ini dalam folder `Mailbox/KRS` dan akan mengarahkan e-mail ke program *autoresponse* KRS. Dimana program *autoresponse* hanya menanggapi request berupa KRS dan membalas e-mail tersebut.

3.1 E-R Diagram

Diagram E-R merupakan suatu diagram yang digunakan untuk memodelkan hubungan antara entitas-entitas yang ada di lingkungan sistem. Tahapan rencana pendahuluan dalam program adalah menentukan lingkup sistem yang akan ditangani. Dengan menganalisa ide dasar diatas, maka dapat ditemukan sejumlah entitas untuk membentuk Diagram E-R, yaitu :

- Mahasiswa
- Sistem Autoresponse
- Mail Reply

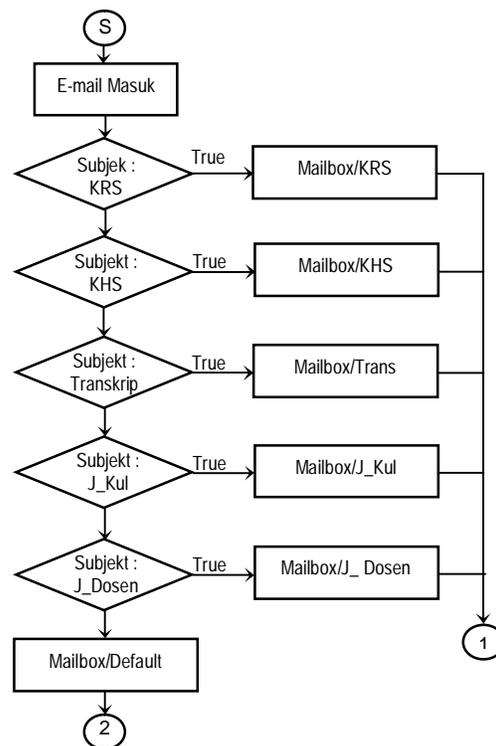


Gambar 3.1 E-R Diagram Sistem *Mail Autoresponse*

Berikut ini adalah gambar E-R Diagram program *mail autoresponder* untuk layanan informasi akademik. Saqlah satu hubungan dari entitas diatas adalah sistem autoresponder merespon dan mengirimkan *mail reply*.

3.2 Flowchart dan Test Case

Flowchart dan algoritma disusun untuk mempermudah dalam pembuatan program, sehingga dapat memperjelas alur kerja program. Dalam sistem *mail autoresponse* ini sebenarnya dapat dibuat 2 flowchart; flowchart yang pertama untuk *mail filtering* yang menggunakan procmail dan yang kedua adalah flowchart program yang merupakan *autoresponse* -nya. Berikut ini adalah flowchart untuk mail filtering yang dilakukan oleh procmail.

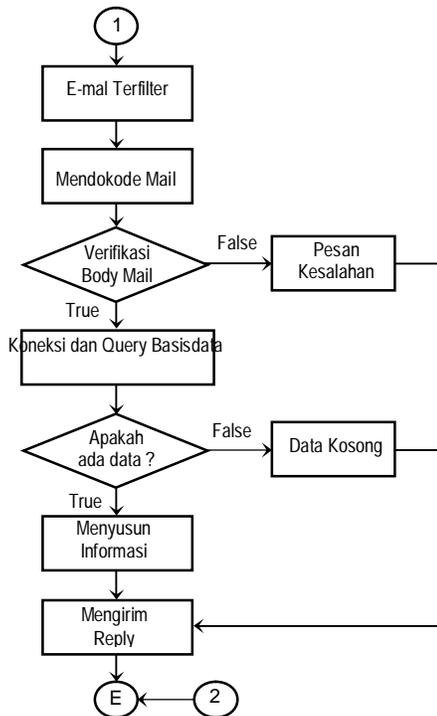


Gambar 3.2 Flowchart *Mail Filtering*

Fungsi dari *mail filtering* untuk menaruh mail request ke dalam folder mailbox sesuai dengan subject mail, karena request ditulis dalam subject e-mail. Selain itu juga berfungsi untuk

mengarahkan e-mail ke program yang sesuai. Algoritma penjelasan flowchartnya adalah sebagai berikut: Request yang masuk berupa e-mail dari pengirim di-filter, jika subjek berupa 'KRS' maka e-mail akan ditaruh dalam folder mailbox/KRS dan e-mail diarahkan ke program *autoresponse* yang menangani request KRS. Proses ini berlangsung hingga subjek tidak cocok dengan satupun kriteria filter, jika tidak cocok maka e-mail cukup ditaruh dalam mailbox/default.

Sedangkan flowchart program *autoresponse* ditunjukkan pada gambar berikut.



Gambar 3.3 Flowchart Mail Autoresponse

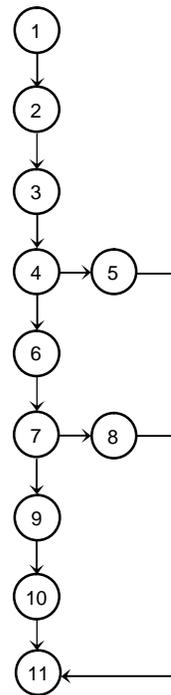
Dalam sistem terdapat beberapa program *autoresponse* yang memiliki mekanisme yang sama. Flowchart diatas dapat dijelaskan dalam langkah-langkah algoritma sebagai berikut:

1. Email yang telah ter-filter akan di-parse oleh fungsi `Mail_Decode` sehingga dihasilkan Header-header dan Body dari e-mail
2. Selanjutnya setelah didapatkan body maka perlu dilakukan pengecekan terhadap format variabel yang ada di dalam body. Jika formatnya benar maka dilanjutkan proses verifikasi terhadap password. Kalau format dan verifikasi tidak cocok maka program akan menghasilkan error message yang nantinya akan dikirim dalam bentuk e-mail.
3. Jika password dari email sesuai maka dilakukan koneksi ke basisdata untuk melakukan query

data. Jika data tidak ditemukan/belum tersedia maka program akan menghasilkan no data message.

4. Selanjutnya data-data yang telah didapat dari basisdata disusun sehingga membentuk informasi yang tadi di-request.
5. Informasi ini selanjutnya dikirim oleh program ke alamat si pengirim tadi sebagai balasan e-mail.

Usulan untuk *test-case* bisa dibuat dengan menentukan path-path yang mungkin ditempuh dalam program. Jumlah path yang mungkin ditempuh dalam sebuah program dapat diketahui dengan menganalisa *flowgraph* dari program tersebut. *Flowgraph* dapat disusun dari flowchart program yang telah dibuat di atas. Setiap blok diganti dengan node dan diberi nomor agar nantinya mudah digunakan untuk menentukan lintasan-lintasannya. Berikut ini adalah *flowgraph* dari program *mail autoresponse*.



Gambar 3.4 Flowgraph Mail Autoresponse

Dari *flowgraph* tersebut, kita dapat menentukan jumlah lintasan (*path*) yang mungkin ditempuh dengan cara menghitung kompleksitas dari *flowgraph* tersebut. Kompleksitas dapat kita hitung dengan tiga cara, namun sebelum menghitung kompleksitas perlu diketahui beberapa hal yaitu region, edge, dan *Predicate Node*. Region adalah jumlah wilayah yang dilalui lintasan dari awal hingga akhir. Edge adalah seluruh jalur atau

yang ditunjukkan dengan anak panah, sedangkan *Predicate Node* adalah titik percabangan.

$$V(G) = \text{Jumlah Region yang ada} = 3 \text{ region}$$

$$V(G) = \text{Jumlah Edge} - \text{Jumlah Node} + 2 \\ = 12 - 11 + 2 = 3$$

$$V(G) = \text{Jumlah Predicate Node} + 1 \\ = 2 + 1 = 3$$

Dari hasil perhitungan tersebut, kita memperoleh nilai kompleksitas $V(G)$ dari *flowgraph* tersebut yaitu 3. Jumlah path yang mungkin ditempuh sama dengan besar nilai kompleksitas dari *flowgraph* tersebut. Berikut ini adalah path-path dan *test-case* yang mungkin ditempuh dalam program adalah :

1. Lintasan: 1-2-3-4-5-11

Dalam path ini, parameter-parameter (NIM, Password atau Kode) hasil dari parse mail terdapat kesalahan format penulisan atau NIM dan Passwordnya tidak cocok sehingga dihasilkan pesan kesalahan yang langsung menjadi isi e-mail balasan.

2. Lintasan: 1-2-3-4-5-6-7-8-11

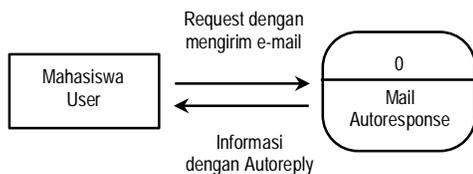
Path ini format dan verifikasi parameter semua cocok, sehingga dilakukan query ke basisdata untuk menyusun informasi. Namun ketika data yang diambil kosong, maka akan dihasilkan pesan bahwa data belum atau tidak ada. Pesan ini menjadi isi dalam e-mail balasan

3. Lintasan: 1-2-3-4-5-6-7-8-9-10-11

Kemudian di path ini, semua verifikasi dan data hasil query tersedia, sehingga data-data tersebut disusun menjadi informasi yang diminta untuk kemudian dikirim dalam bentuk e-mail sebagai balasan

3.3 Diagram Kontek dan Data Flow Diagram

Untuk memperjelas gambaran dari sistem *mail autoreponse* perlu dibuat diagram konteks yang nantinya dapat dikembangkan untuk menyusun DFD lebih detail. Gambar 3.3 adalah diagram konteks yang menunjukkan sistem dasar dari *mail autoreponse*.



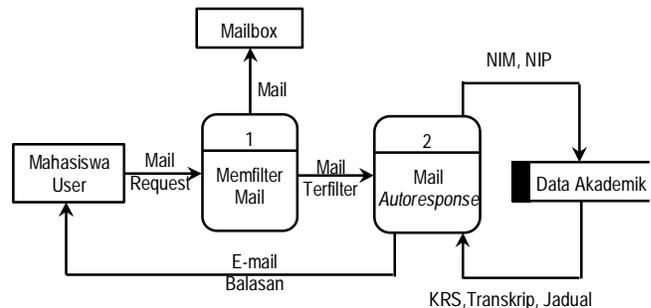
Gambar 3.5 Diagram Kontek dari Sistem Mail Autoreponse

Mahasiswa sebagai user yang menginginkan informasi akademik dapat mengirimkan request dalam bentuk e-mail ke mail server (alamat e-mail) SIA. Kemudian sistem *autoreponse* yang terdapat dalam SIA akan membalas informasi yang diinginkan secara otomatis (*autoreply*)

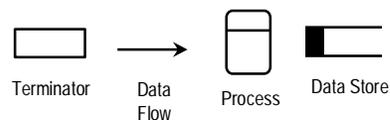
DFD merupakan alat yang biasa dipakai untuk mendokumentasikan proses dalam sistem. Selanjutnya menekankan pada fungsi-fungsi di dalam sistem, cara menggunakan informasi yang tersimpan dan pemindahan informasi antar fungsi di dalam sistem.

DFD memperlihatkan sistem sebagai proses, tujuannya adalah memberikan pandangan umum dari sistem. Proses yang diperlihatkan berinteraksi dengan lingkungannya. Ada pihak luar atau lingkungan yang memberi masukan dan ada pihak yang menerima keluran sistem. Pihak luar (*terminator/external entity*) dapat berupa sistem lain perangkat keras, orang atau organisasi, dalam Sistem *Autoreponse* ini yang bertindak sebagai terminator adalah mahasiswa sebagai user yang mengirim request dan mailbox untuk tempat menyimpan email yang masuk.

Pada tahap perencanaan selanjutnya, DFD akan dijabarkan ke pandangan yang lebih detail. Gambar 3.6 memperlihatkan DFD level satu yang merupakan penjabaran dari diagram konteks sebelumnya.

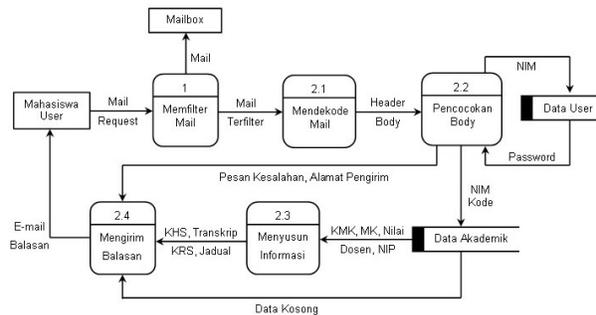


Keterangan:



Gambar 3.6 DFD Level 1 Sistem Mail Autoreponse

Terlihat pada DFD level satu di atas sistem *autoreponse* terbagi menjadi dua proses utama yaitu mem-filter e-mail dan proses *autoreponse* e-mail yang sudah di-filter berdasarkan subjeknya. Proses mail *autoreponse* berfungsi melakukan tanggapan secara otomatis terhadap permintaan yang dikirim, kemudian menyusun informasi dan mengirimkan balasan ke pengirim e-mail tadi. Dalam proses mail *autoreponse* terbagi lagi menjadi beberapa proses, yang dapat dilihat pada DFD level 2 gambar 3.7.



Gambar 3.7 DFD Level 2 Sistem Mail Autoresponse

Pada DFD sistem *mail autoresponse* tampak bahwa terdapat 5 proses yang terjadi. Proses pertama yaitu memfilter e-mail yang masuk, jika sesuai dengan kriteria maka e-mail diarahkan ke program untuk dilakukan proses decoding fungsinya untuk mem-parse e-mail sehingga didapat *header-header* dan *body*. *Body* mail hasil *decoding* kemudian diproses dalam pencocokan format maupun verifikasi. Verifikasi berguna untuk mencocokkan NIM dan Password yang dicocokkan dari *body* dan basisdata. Data alamat pengirim diumpunka ke proses pengiriman e-mail. Selanjutnya dilakukan query ke basisdata dengan NIM atau kode sesuai request yang diminta dalam e-mail tadi. Data yang diambil dari basisdata kemudian disusun untuk membentuk suatu informasi yang diinginkan. Kemudian informasi dikirim ke si pengirim *mail request*.

IV. Implementasi dan Pengujian Sistem

Perancangan sistem yang telah dijabarkan dalam bab III selanjutnya akan diimplementasikan dalam bentuk kode-kode program (*source code*) menggunakan bahasa pemrograman PHP. Dimana PHP disini adalah PHP dalam mode biner (`/usr/bin/php`) yang akan berjalan atas input dari *mail filter*, bukan mode modul Apache, karena tidak mengeksekusi skrip PHP via *browser*. Skrip PHP ditempatkan dalam *mail server* atau *mail address* yang merupakan alamat tujuan user dalam mengirimkan *mail request*.

Pengujian terhadap sistem yang sudah dibangun adalah dengan mengirim e-mail ke alamat yang sudah ditentukan dimana didalamnya terdapat *system autoresponse*. E-mail akan menjadi input bagi sistem, kemudian mengamati respon sistem hingga diperoleh kesesuaian antara *mail request* dengan balasan e-mail dari sistem yang sesuai dari tujuan pembangunan sistem.

Sistem *mail autoresponse* ini digunakan untuk menangani enam *mail request* yang berupa Jadwal Kuliah, Jadwal Ujian, Jadwal Dosen, Kartu Rencana Studi (KRS), Kartu Hasil Studi (KHS) dan Transkrip Nilai. Sehingga implementasi untuk pembuatan kode program PHP juga terdapat enam

program. Artinya satu program PHP hanya akan menangani satu *mail request*, misalnya *mail request* dengan subjek KRS akan ditangani oleh program PHP dengan nama file `krs.php`. Disamping pembuatan kode program PHP, juga yang tidak kalah penting adalah pembuatan skrip untuk *mail filter* (`.procmailrc`). Sebab dengan *mail filter* ini program PHP akan menerima *trigger* e-mail sehingga program dapat berjalan.

4.1 Implementasi Kode Program Mail Filter

Pertama-tama yang dapat kita lakukan adalah membuat kode program *mail filter*. Dengan `procmail` kita dapat membangun mail filter dengan membuat file `.procmailrc` dan sebuah folder `procmail` yang berisi file `log`. Folder `procmail` ini ditempatkan pada direktori `/home/dicky/`. File `.procmailrc` cukup ditaruh dalam direktori `/home/dicky` sama dengan kode-kode program PHP lainnya yang akan dibuat. *Mail filter* disini mempunyai fungsi sebagai berikut:

1. Untuk meletakkan *mail request* yang masuk ke alamat e-mail/mail server (`dicky@localhost.localdomain`) ke dalam folder-folder sesuai subjek requestnya dan
2. Mengumpulkan *mail request* ke program PHP sesuai dengan subjek requestnya. Artinya e-mail tadi menjadi input dari program PHP sesuai isi subjeknya.

Skrip yang digunakan dalam file `.procmailrc` adalah sebagai berikut :

```

PATH=/usr//bin:/bin:/usr/bin::/usr/local/bin:/usr/X11R6/bin:/usr/games:/home/dicky/bin
SHELL=/bin/sh
MAILDIR=$HOME/Mail
DEFAULT=$HOME/Mail/inbox/
PMDIR=$HOME/.procmail
LOGFILE=$PMDIR/log
VERBOSE=yes
LOGABSTRACT=all
AUTOREPLYDIR=$HOME

SUBJEK=`formail -zxSubjek:`

:0
*! ^X-Loop: dicky@localhost\.localdomain
{
#Request KHS
:0c:
* ^Subjek:.*khs
khs/
:0
* ^Subjek:.*khs
| $HOME/khs.php

```

Dengan skrip diatas e-mail dengan subjek KHS yang masuk ke alamat `dicky@localhost.localdomain` akan ditaruh dalam folder `/mailbox/khs` dan selanjutnya dalam waktu yang berurutan mengumpulkan e-mail ke program `khs.php`. Namun jika e-mail yang masuk tidak

memiliki subjek yang berisi khs, maka akan diteruskan ke skrip berikutnya/dibawahnya.

4.2 Implementasi Kode Program PHP

Selanjutnya adalah implementasi kode program PHP, di dalam program ini barulah e-mail akan diproses untuk mendapatkan informasi dan proses *autoreply* berlangsung disini. Semua kode program PHP memiliki fungsi untuk mengolah *mail request* sesuai request yang diminta dan menyusun informasinya, kemudian mengirimkan balasan e-mail tersebut. Sehingga semua program PHP harus dapat menyelesaikan tugasnya untuk mendekode e-mail hasil *filter* dari procmail. Pekerjaan ini dapat dilakukan dengan fungsi yang mengungkus paket *Mail_Mime*, fungsi paket ini adalah untuk mendekode e-mail sehingga bisa dipecah menurut *header-header* dan *body message*.

Kesemua kode program PHP mempunyai skrip untuk mendekode e-mail dengan kelas *Mail_mimeDecode* yang masih merupakan bagian paket *PEAR Mail_Mime*. Skrip dimulai dengan membaca e-mail hasil filter dari procmail yang diarahkan ke program PHP. Selanjutnya mendekode mail menggunakan kelas *Mail_mimeDecode::(\$params)*, hasilnya ddisimoan dalam sebuah array dan apabila dilihat strukturnya akan berupa:

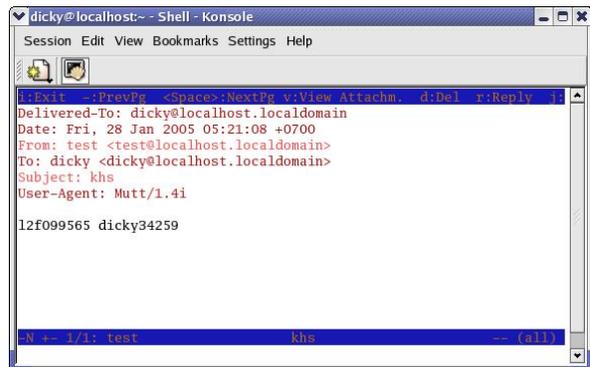
```
stdClass Object
(
    [headers] => Array
    (
        [received] => by localhost
(Postfix, from userid 500) id 22F5920628;
Wed, 19 Jan 2005 01:34:25 -0500 (EST)
        [date] => Wed, 19 Jan 2005
13:34:25 +0700
        [from] => mailuser
<mailuser@localhost.localdomain>
        [to] =>
dicky@localhost.localhost
        [subjek] => khs
        [message-id] =>
<20050119063425.GA5532@localhost.localdoma
in>
        [mime-version] => 1.0
        [content-type] => text/plain;
charset=us-ascii
        [content-disposition] =>
inline
        [user-agent] => Mutt/1.4i
    )
    [ctype_primary] => text
    [ctype_secondary] => plain
    [ctype_parameters] => Array
    (
        [charset] => us-ascii
    )
    [disposition] => inline
    [body] => L2F099565 dicky34259
)
```

Selanjutnya informasi yang ada di masing-masing *header* dan *body* dapat mudah disimpan ke dalam sebuah variabel-variabel global. Informasi *body message* disimpan dalam variabel *\$body*, dan informasi alamat pengirim e-mail disimpan dalam variabel *\$sender*.

Dalam proses verifikasi *body message* digunakan bantuan *Reguler Expression (Regex)*, pencocokan password dilakukan dengan enkripsi MD5. Sedangkan dalam melakukan proses pengiriman e-mail kepada pengirim request digunakan fungsi *mail()*.

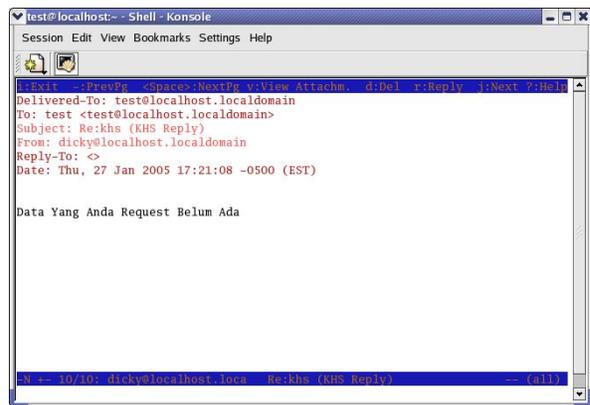
4.3 Pengujian Mail Request KHS

Dalam pengujian, dilakukan dengan mengirim e-mail menggunakan alamat *dicky@localhost.localdomain*. Subjek e-mail (*subject*) diisi dengan **KHS**. Dan isi nya (*body*) ditulis dengan format *NIM<spasi>password*. Sehingga jika diterima di mail server bentuk e-mailnya adalah sebagai berikut:



Gambar 4.1 Mail Request KHS

Balasan e-mail dari mail server tentunya adalah informasi KRS yang mempunyai NIM: *L2F099565*, karena melihat penulisan format NIM dan password sudah benar. Hasil e-mail balasan yang ada di dalam mailbox user adalah sebagai berikut:

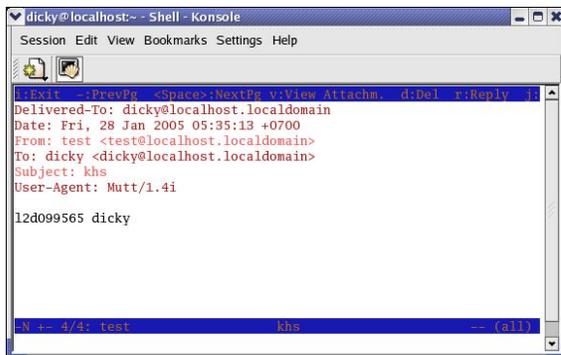


Gambar 4.2 E-mail Reply KHS Data Kosong

Namun terlihat dari gambar diatas bahwa pesan didalamnya adalah "Data Yang Anda Request Belum Ada". Hal ini menunjukkan bahwa mahasiswa dengan NIM L2F099565 belum melakukan pengisian KRS. Sehingga datanya belum ada dalam basisdata akademik.

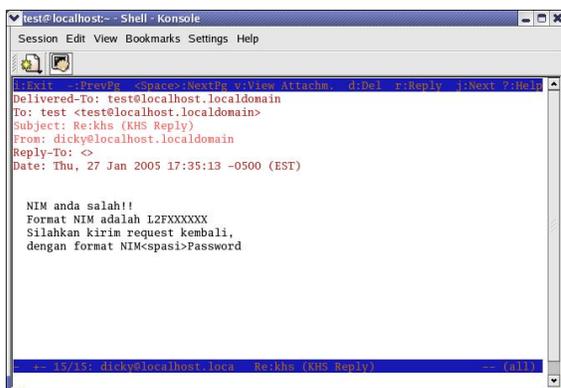
Dalam mengirim *mail request* dimungkinkan terjadi kesalahan dalam penulisan, jika terjadi kesalahan dalam penulisan subjek maka sistem tidak dapat mendeteksi, sehingga e-mail tidak akan terbalas, e-mail tersebut secara *default* hanya akan diletakkan dalam *inbox*. Namun jika terjadi kesalahan dalam *body message*, sistem akan mendeteksi dan dapat membalas berupa pesan kesalahan. Kesalahan tersebut adalah :

1. Kesalahan penulisan NIM yang dapat ditunjukkan dalam gambar 4.3 yang seharusnya format NIM sesuai gambar 4.2.



Gambar 4.3 NIM salah dalam Mail Request

Balasan yang didapat di *mailbox* user adalah sebagai berikut:



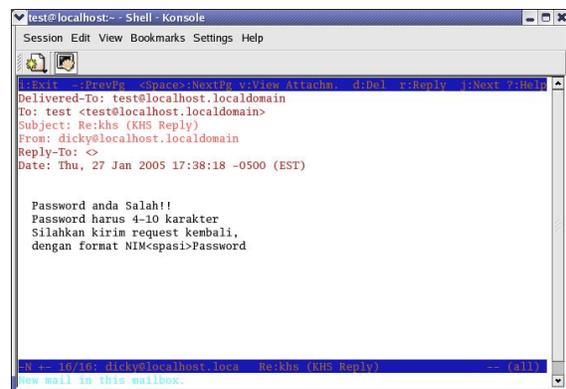
Gambar 4.4 Pesan Kesalahan NIM dalam E-mail Reply

2. Kesalahan penulisan NIM yang dapat ditunjukkan dalam gambar 4.5 yang seharusnya format password adalah karakter dengan panjang 4-10



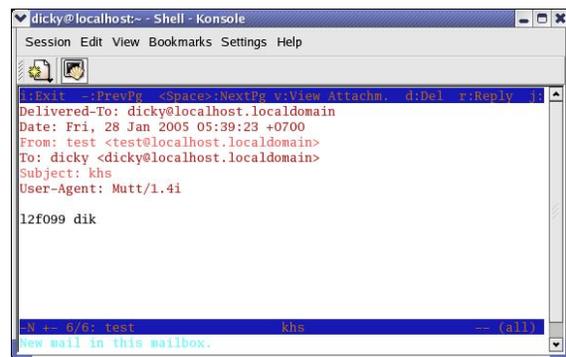
Gambar 4.5 Password salah dalam Mail Request

Balasan yang didapat di *mailbox* user adalah sebagai berikut:



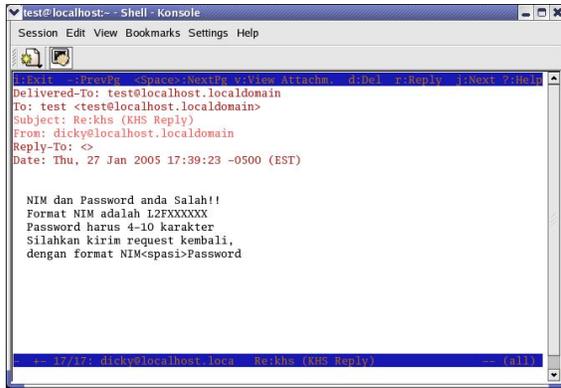
Gambar 4.6 Pesan Kesalahan Password dalam E-mail Reply

3. Kesalahan penulisan NIM sekaligus password yang dapat ditunjukkan dalam gambar 4.7



Gambar 4.7 NIM dan Password Salah dalam Mail Request

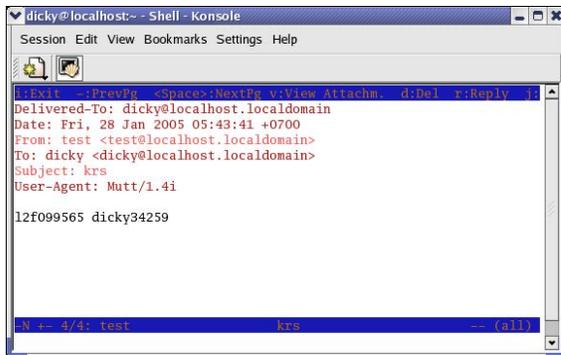
Balasan yang didapat di *mailbox* user seperti yang terlihat pada gambar 4.8.



Gambar 4.8 Pesan Kesalahan NIM dan Password dalam E-mail Reply

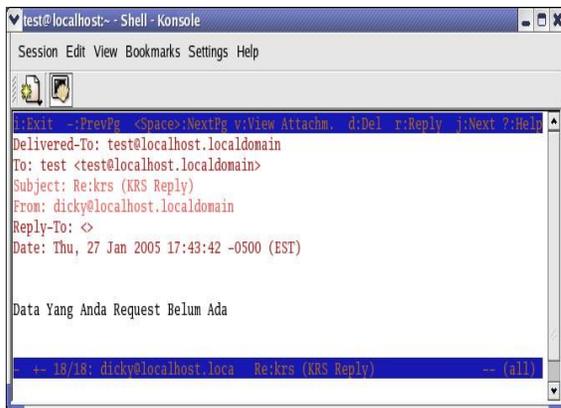
4.4 Pengujian Mail Request KHS

Pengujian dilakukan dengan mengirimkan e-mail dari user dengan mengisi baris subjek dengan **KHS** dan menulis NIM dan password dengan benar pada *body message*, seperti ditunjukkan pada gambar 4.9, yaitu e-mail yang diterima dalam mailbox server (/mailbox/krs/).



Gambar 4.9 Mail Request KRS

Sehingga balasan yang didapat dalam mailbox user adalah sebagai berikut:

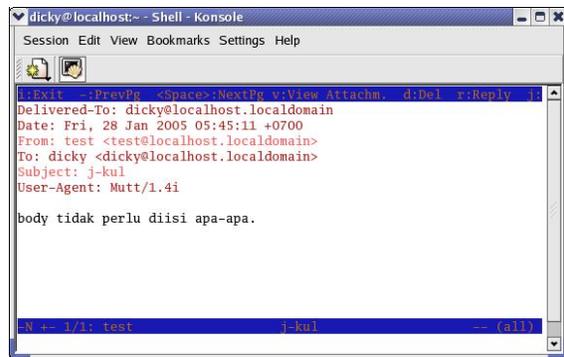


Gambar 4.10 E-mail Reply KRS Data Kosong

Hal ini terjadi karena mahasiswa belum mengisi KRS sehingga data KHS juga otomatis kosong. Artinya bahwa mahasiswa dengan NIM L2F099565 belum mempunyai data KHS dalam basidata akademik. Sedangkan kesalahan yang mungkin terjadi dalam request KRS sama dengan request KHS.

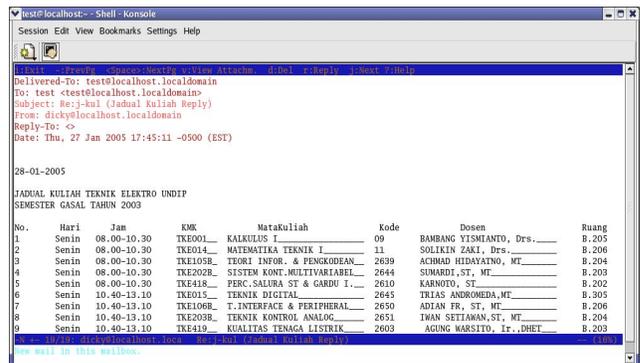
4.5 Pengujian Mail Request Jadwal Kuliah

Memulai pengujian dengan mengirim mail request ke “dicky”. Dalam baris subjek diisi dengan **j-kul**, sedangkan dalam *body message* tidak perlu diisi apa-apa, karena dalam request Jadwal Kuliah memang tidak dilakukan proses verifikasi. E-mail yang dikirim akan masuk dalam folder /Mailbox/j-kul/, bentuknya seperti ditunjukkan pada gambar 4.11.



Gambar 4.11 Mail Request Jadwal Kuliah

Hasilnya adalah mail server secara otomatis membalas e-mail berupa informasi Jadwal Kuliah ke user. E-mail yang diterima dalam mailbox user ditunjukkan pada gambar 4.12.



Gambar 4.12 E-mail Reply Jadwal Kuliah

4.6 Pengujian Mail Request Jadwal Dosen

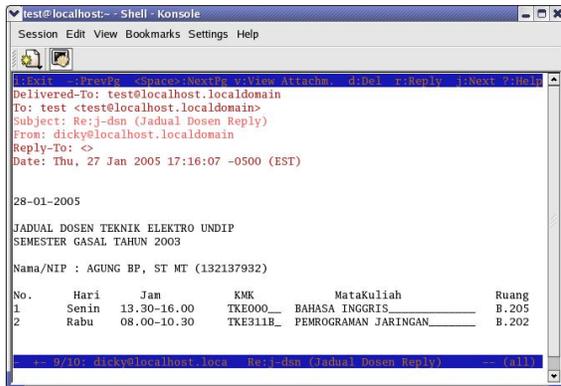
Memulai pengujian dan mengirim mail request ke “dicky” dengan mengisi subjek dengan **j-dsn**, *body message* harus diisi dengan kode dosen

yang terdiri dari empat digit angka, seperti gambar 4.13.



Gambar 4.13 Mail Request Jadual Dosen

Hasilnya adalah mail server secara otomatis membalas e-mail berupa informasi jadual dosen yang bersangkutan ke user, e-mail tersebut adalah sebagai berikut:

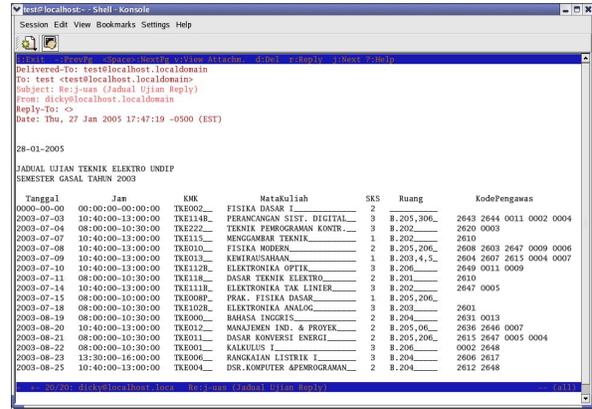


Gambar 4.14 E-mail Reply Jadual Dosen

Dalam penulisan mail request dimungkinkan terdapat kesalahan penulisan kode dosen, jika hal ini terjadi maka sistem akan membalas e-mail yang berisi pesan kesalahan penulisan kode dosen.

4.7 Pengujian Mail Request Jadual Ujian

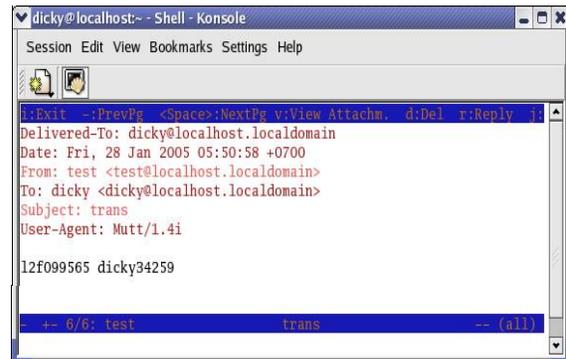
Dalam pengujian request Jadual Ujian sama ketika dilakukan pengujian Jadual Kuliah, hanya saja yang berbeda adalah penulisan **j-uas** pada subjek e-mail. Sedangkan e-mail balasan akan berupa informasi Jadual Ujian seperti gambar 4.15.



Gambar 4.15 E-mail Reply Jadual Ujian

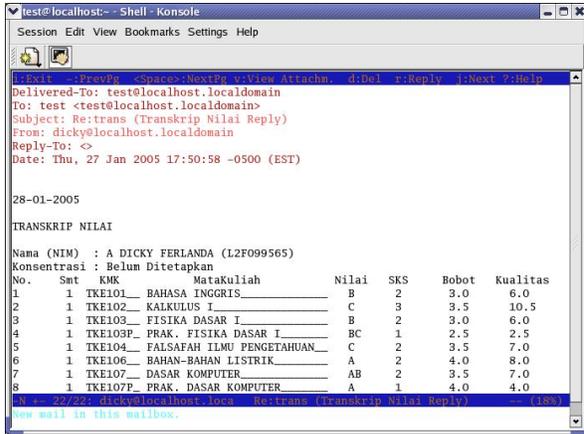
4.8 Pengujian Mail Request Transkrip Nilai

Pengujian dimulai dengan mengirim e-mail ke "dicky", mengisi baris subjek dengan **trans**. Dalam body message perlu diisikan NIM dan password untuk keperluan verifikasi, dengan format penulisan NIM<spasi>password. Gambar 4.20 menunjukkan pengiriman mail request Transkrip Nilai yang benar dan diambil dari mailbox "dicky" yang berada dalam folder /Mail/trans.



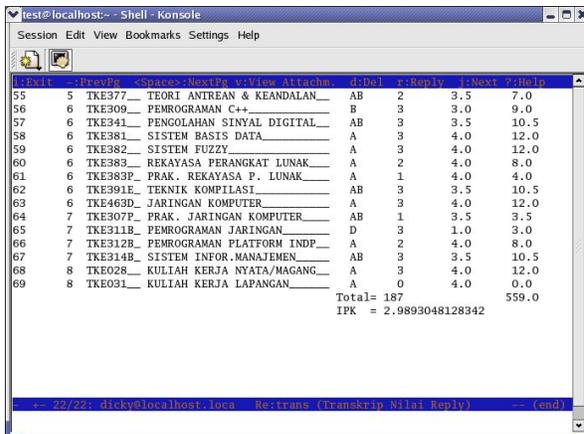
Gambar 4.16 Mail Request Transkrip Nilai

Dengan menerima mail request, sistem akan secara langsung merespon sesuai dengan requestnya yang tertulis di subjek dan memberi balasan e-mail seperti gambar 4.17 dan 4.18.



Gambar 4.17 E-mail Reply Transkrip Nilai (bagian atas)

Gambar 4.21 menunjukkan e-mail balasan Transkrip Nilai bagian atas, sedangkan gambar 4.22 menunjukkan e-mail balasan bagian bawah.



Gambar 4.18 E-mail Reply Transkrip Nilai (bagian bawah)

Kedua gambar diatas adalah satu e-mail balasan dari mail server ("dicky"), yang diambil dari mailbox user. Tampak bahwa informasi Transkrip Nilai terdapat Total SKS, Bobot Nilai sekaligus IPK.

Dalam mail request Transkrip Nilai juga mungkin terdapat kesalahan dalam menulis format NIM<spasi>Password. Jika hal ini terjadi mak sistem akan memberikan balasan e-mail yang

berisi pesan kesalahan, seperti yang terdapat dalam pengujian mail request KHS.

V. Kesimpulan dan Pengujian Sistem

5.1 Kesimpulan

Setelah melalui tahapan perancangan program, implementasi dan pengujian Sistem *Autoreponse*, maka diperoleh hasil-hasil yang dapat diambil beberapa kesimpulan yaitu:

1. Bahasa pemrograman yang dipakai dalam membangun sistem adalah PHP dengan *mode biner*, karena program berjalan atas input dari e-mail request bukan dari request dari *browser client*.
2. E-mail request yang merupakan input bagi skrip PHP (*mode biner*) adalah e-mail yang telah terfilter oleh *procmail*, berdasarkan subjeknya.
3. *Sistem autoreponse* dapat diterapkan dalam Layanan Informasi akademik via E-mail. Sedangkan Layanan yang dapat ditangani berupa *mail request* KHS, KRS, Jadwal Kuliah, Jadwal dosen, Jadwal Ujian dan Transkrip Nilai.
4. Untuk mendapatkan informasi *headers* dan *body* dari e-mail, program PHP menggunakan sebuah paket dari PEAR yaitu *Mail_Mime*, yang berfungsi untuk mendekode e-mail.
5. *Sistem autoreponse* juga dapat mendeteksi adanya kesalahan dalam penulisan dalam *body message* dan mengirimkan balasan e-mail berupa pesan kesalahan yang terjadi.

5.1 Saran

Setelah dilakukan pengujian pada *sistem autoreponse* yang telah diimplementasikan untuk layanan inormasi akademik, dapat diberikan beberapa saran sebagai berikut:

1. Layanan dapat dikembangkan menjadi mail yang dapat menginput ke sistem informasi akademik (basisdatanya) tidak hanya berupa mail request.
2. Informasi dalam e-mail balasan masih berupa *text mode*, selanjutnya dapat dikembangkan *e-mail attachment* yang berisi data yang *comaptible* dengan Ms. Excel. Sehingga akan memudahkan mahasiswa dalam mengedit informai akademik.
3. Karen sistem ini mengambil data dari basisdata akademik, maka hendaknya basisdata sistem akademik selalu terbaru. Sehingga dapat menghasilkan informasi yang aktual pula.

VI. Daftar Pustaka

1. Azis, M. Farid, “*Belajar Sendiri Pemrograman PHP 4*”, PT Elex Media Komputindo, Jakarta 2001.
2. Dennis, Jim, “*Procmal Mini-Tutorial: Automated Mail Handling*”, <http://www.linuxgazette.com/issue14/procmal.html>
3. Hakim, Lukmanul dan Musalini, Uus, “*150 Rahasia dan Trik Menguasai PHP*”, PT Elex Media Komputindo, Jakarta 2003.
4. Haryanto, Steven, “*PHP: Kumpulan Resep Pemrograman*”, PT Dian Rakyat, Jakarta, 2004.
5. Haryanto, Steven, “*Regex: Kumpulan Resep Pemrograman*”, PT Dian Rakyat, Jakarta, 2004.
6. Plew, Ronald R. and Stephens, Ryan K., “*SAMS Teach Yourself SQL in 24 Hours*”, Sams Publishing, Indianapolis – Indiana, USA, 2000.
7. Sakti, Nufansa Wira, “*Belajar Sendiri Menggunakan SQL*Plus*”, PT Elex Media Komputindo, Jakarta 1999.
8. Salmi, Timo, “*Timo's procmal tips and recipes*”, <http://www.uwasa.fi/~ts/info/proctips.html>, 2003.
9. Soboroff, Ian, “*Mail Filtering with Procmal*”, <http://userpages.umbc.edu/~ian/procmal.html>, 1997
10. The PEAR Documentation Group, “*PEAR Manual*”, 2004.
11. The PHP Documentation Group, “*PHP Manual*”, 2004.
12. Wankyu Choi, Allan Kent, Chris Lea, Ganesh Prasad, Chris Ullman, “*Beginning PHP4*”, Wrox Press Ltd, Birmingham, UK, 2000.
13. Welling, Luke and Thomson, Laura, “*PHP dan MySQL Web Development*”, Sams Publishing, Indianapolis – Indiana, USA, 2003.
14. -----, “*Procmal FAQ*”, <http://rhols66.adsl.netsonic.fi/era/procmal/mini-faq.html>
15. -----, “*Procmal*”, <http://www.udel.edu/topics/e-mail/procmal/procmal.html>

A. Dicky Ferlanda adalah mahasiswa Teknik Elektro Konsentrasi Komputer dan Informatika. Saat ini sedang menjalani proses untuk menyelesaikan studi Strata I pada Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang.

Mengetahui/Menyetujui

Pembimbing I

Pembimbing II

Ir Kodrat I.S, MT
NIP. 132 046 696

Agung BP, ST MIT
NIP. 132 137 932