



# PETIR

## JURNAL PENGKAJIAN DAN PENERAPAN TEKNIK INFORMATIKA

VOLUME 8 - NOMOR 2

SEPTEMBER 2015

ISSN 1978-9262

IMPLEMENTASI DAN PENGUJIAN KINERJA ORACLE 10g *REAL APPLICATION CLUSTER* (RAC) PADA SISTEM OPERASI SUN SOLARIS 10

*Gatot Budi Santoso; Yanuar Indra Wirawan*

RANCANG BANGUN APLIKASI *MONITORING* PENCADANGAN DAYA LISTRIK DENGAN MEMANFAATKAN TENAGA KINCIR ANGIN

*Meilia Nur Indah Susanti*

APLIKASI PENGOLAHAN DATA PASIEN, STUDI KASUS RSUD SAWERIGADING PALOPO SULAWESI SELATAN

*Abdul Haris; Alan Burhan*

PENGUNAAN JARINGAN SYARAF TIRUAN DENGAN METODE BACKPROPAGATION DALAM MEMPREDIKSI INDEKS HARGA SAHAM GABUNGAN (IHSG)

*Wisnu Hendro Martono; Dian Hartanti*

APLIKASI KURSUS KOMPUTER *ONLINE* MENGGUNAKAN PHP PADA LEMBAGA KURSUS KOMPUTER YOGZ COURSE

*Harni Kusniyati; Yoga Hapsara Mursidigama*

MONITORING AKSES LOKER DOSEN MENGGUNAKAN *EMBEDDED SYSTEM* DENGAN ANTARMUKA ANDROID

*Riki Ruli A. Siregar; Jaka Mahardika*

TATA KELOLA TINGKAT LAYANAN SISTEM INFORMASI PEMESANAN TIKET MENGGUNAKAN KERANGKA KERJA COBIT 4.1 PADA ARNES SHUTTLE CABANG KOTA BANDUNG

*R.Fenny Syafariani; Gilang Nandapratama*

PERANCANGAN APLIKASI SISTEM PENDUKUNG KEPUTUSAN BERBASIS WEB UNTUK MENENTUKAN PENJURUSAN PADA SMA X DENGAN MENGGUNAKAN METODE AHP (*ANALYTICAL HIERARCHY PROCESS*)

*Yasni Djamain*

IMPLEMENTASI DEMPSTER SHAFER DALAM MENGHASILKAN KEPUTUSAN PENGAMBILAN TOPIK TUGAS AKHIR BAGI MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA UMB

*Desi Ramayanti*

SISTEM LAPORAN KEUANGAN DENGAN MENGGUNAKAN MOBILE PHONE, PHP DAN MYSQL


*Marliana Sari*

SISTEM MONITORING LABORATORIUM KOMPUTER PUSAT UNIVERSITAS MERCU BUANA DENGAN MENGGUNAKAN METODE SCREEN THIEF

*Sarwati Rahayu*

APLIKASI ANTRIAN SMS MENGGUNAKAN *MULTIPLE CHANNEL* DAN *MULTI PHASE SISTEM* DI PT IVM (INTITEK VIRTULINDO MANDIRI)

*Raka Yusuf; Harni Kusniyati; Yuyus Mohayus*

 ISSN 1978-9262 771978 926272	SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)					
	PETIR	VOL. 8	NO. 2	HAL. 133 - 239	JAKARTA, SEPTEMBER 2015	ISSN 1978-9262



# APLIKASI ANTRIAN SMS MENGGUNAKAN *MULTIPLE CHANNEL* DAN *MULTI PHASE* SISTEM DI PT IVM (INTITEK VIRTULINDO MANDIRI)

Raka Yusuf<sup>1</sup>, Harni Kusniyati<sup>2</sup>, Yuyus Mohayus<sup>3</sup>

E-mail: rakayusuf@yahoo.co.id<sup>1</sup>, harni246@gmail.com<sup>2</sup>, mohayus@gmail.com<sup>3</sup>

## Abstrak

*Content Provider* adalah perusahaan yang bergerak dalam bisnis telekomunikasi dengan jasa pelayanan SMS premium. Perkembangan dunia telekomunikasi dan teknologi membuat *content provider* saling bersaing untuk membuat layanan atau konten yang menarik. Dengan banyaknya pelanggan yang menggunakan konten tersebut maka permintaan terhadap layanan SMS meningkat.

PT IVM adalah salah satu perusahaan *content provider*. Aplikasi SMS Gateway masih berjalan secara sinkron. Hal ini menyebabkan permintaan pelanggan tidak dapat dilayani dengan baik. Oleh karena itu dibutuhkan suatu aplikasi antrian yang dapat menampung data SMS, sebelum data diproses dan dikirim ke operator.

Aplikasi antrian yang dibuat menggunakan banyak *channel* (*Multiple Channel*) dan banyak *phase* (*Multi Phase*). Aplikasi dibagi berdasarkan operator dan terdiri dari 3 *phase*, antrian proses, antrian kirim dan antrian basis data. Antrian proses adalah antrian yang menampung data yang masuk dari operator. Antrian kirim adalah antrian yang menampung data yang sudah diproses dan siap dikirim ke operator. Dan antrian basis data adalah antrian yang menampung data untuk dimasukkan ke dalam basis data sebagai log transaksi.

**Kata Kunci :** *Content Provider*, SMS, antrian, *Multiple Channel*, *Multi Phase*, operator.

## 1. PENDAHULUAN

SMS (*Short Message Service*) merupakan layanan pesan yang pada umumnya terdapat pada setiap sistem jaringan *wireless digital* (jaringan tanpa kabel). SMS adalah layanan untuk mengirim dan menerima pesan tertulis dari maupun kepada *mobile device* (perangkat bergerak). Pesan teks yang dimaksud tersusun dari huruf, angka, atau karakter alfanumerik.

Perkembangan dunia telekomunikasi dan teknologi membuat *content provider* saling bersaing untuk membuat layanan atau konten yang menarik. Hal ini memicu permintaan user terhadap layanan SMS premium meningkat. Aplikasi yang ada berjalan secara sinkron, mulai dari SMS MO (*Mobile Originated*) tiba sampai SMS MT (*Mobile Terminated*) dikirim ke operator.

Meningkatnya permintaan SMS dan aplikasi yang berjalan secara sinkron, menyebabkan permintaan user tidak dapat dilayani dengan baik. Banyak data yang hilang selama proses SMS dan perbaikan aplikasi (pemroses SMS atau basis data).

## 2. LANDASAN TEORI

### A. SMS (*Short Message Service*)

Merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless*), memungkinkan dilakukannya pengiriman pesan dalam bentuk alfanumerik antara terminal pelanggan atau antar terminal pelanggan dengan sistem eksternal seperti *email*, *voicemail* dan lain-lain.

Isu SMS pertama kali muncul di belahan Eropa sekitar tahun 1991 bersamaan dengan sebuah teknologi *wireless* yang saat ini cukup banyak penggunaannya, yaitu *Global System for Mobile Communication* [2].

### B. Teori Antrian

Teori antrian merupakan suatu cabang dari teori probabilitas terapan, dimana aplikasinya mencakup bidang yang berbeda seperti jaringan komunikasi, sistem komputer dan lain-lain.

Model antrian menjelaskan fenomena yang terjadi dalam kenyataan. Antrian dapat diamati hampir di mana-mana. Kita tahu

tentang hal itu dari kehidupan kita sehari-hari: kita berbaris di depan maskapai penerbangan untuk *check-in*, di depan mesin kopi, di dokter gigi, di perlintasan lalu lintas dan lain-lain. Dalam semua kasus ini, antrian terjadi karena pola kedatangan pelanggan bervariasi dalam waktu dan karakteristik layanan bervariasi dari pelanggan untuk pelanggan.

Dalam rangka menyeragamkan antrian dalam cara yang jelas, notasi Kendall sering digunakan; itu terdiri dari 6 pengidentifikasi, dipisahkan oleh garis vertikal, sebagai berikut [1]:

*Arrivals | Services | Servers | Buffersize | Population | Scheduling*

Dimana "*Arrivals*" mencirikan proses kedatangan pelanggan, "*Services*" persyaratan layanan pelanggan, "*Servers*" jumlah entitas penyedia jasa, "*BufferSize*" jumlah maksimum pelanggan di stasiun antrian, "*Population*" ukuran populasi pelanggan dan akhirnya "*Scheduling*" strategi penjadwalan yang digunakan. Seringkali, ukuran "*Buffersize*" dan "*Population*" dihilangkan dari deskripsi, dalam kasus bahwa mereka dianggap besar tak berhingga. Strategi Penjadwalan juga sering dihilangkan; dalam kasus itu, diasumsikan FCFS. Parameter, terutama "*Arrivals*" dan "*Services*", mungkin memiliki banyak nilai yang berbeda.

Sebagai contoh, M/M/1/8/FCFS merupakan sistem antrian di mana pelanggan tiba menurut proses *Poisson* dan permintaan layanan terdistribusi eksponensial. Sistem tersebut hanya memiliki satu server, sebuah antrian menunggu yang *infinite* dan pelanggan disajikan secara FCFS. Dalam banyak situasi, kita hanya menggunakan tiga parameter pertama, misalnya, M/D/1. Default nilai untuk dua parameter terakhir adalah  $\infty$  dan FCFS [4].

Sistem pelayanan pada umumnya dibagi berdasarkan jumlah jalur atau jumlah server/pelayan dan tahapannya atau jumlah pemberhentian untuk pelayanan [5]:

#### 1. Jumlah ServiceChannel

##### a) Single channel sistem

Pada *single channel* sistem, dengan satu server, dicontohkan dengan restaurant drive trough atau antrian tiket bioskop, dimana antrian dilayani oleh satu pelayan dan dengan satu pemberhentian.

##### b) Multiple channel sistem

*Multiple channel* sistem dapat kita ambil sebagai contoh antrian di bank dimana

antrian dilayani oleh beberapa server (teller). Dalam model ini, lebih dari satu server diasumsikan untuk memberikan layanan. Setiap stasiun layanan diasumsikan untuk memberikan layanan yang sama dan dilengkapi dengan fasilitas serupa untuk layanan. Jarak menunggu menjadi lebih pendek, pada masing-masing stasiun layanan.

#### 2. Jumlah Pemberhentian

##### a) Single phase sistem

*Single phase* sistem, adalah konsumen mendapatkan pelayanan dari satu pemberhentian dan kemudian keluar dari antrian, contohnya adalah pada restaurant fast food (McDonalds, KFC) dimana pelayan yang menerima order memberikan pesanan, sekaligus, menerima pembayaran.

##### b) Multi phase sistem

*Multi phase* sistem, dapat diambil contoh pelayanan perpanjangan STNK kendaraan bermotor, dimana tahap pertama konsumen berada dalam antrian cek fisik kendaraan, setelah selesai konsumen masuk ke antrian kedua untuk melakukan pembayaran pajak, dan pada antrian ketiga menerima STNK yang telah selesai.

#### C. Build and Fix

Model yang paling sederhana dari pengembangan perangkat lunak. Produk ini dibangun dengan persyaratan minimal yang umumnya tidak ada spesifikasi maupun usaha desain dan pengujian yang paling sering diabaikan. Ini adalah representasi dari apa yang terjadi dalam banyak proyek-proyek pengembangan perangkat lunak.

Sayangnya, banyak produk perangkat lunak dikembangkan dengan *build and fix* model. Tanpa spesifikasi atau upaya apapun dalam desain, hanya membangun sebuah produk, dan ulang sebanyak kali diperlukan untuk memuaskan pelanggan. Dan tidak memuaskan untuk ukuran perangkat lunak pengembangan, lebih baik kita menentukan berbagai tahapan proses perangkat lunak [3].

### 3. ANALISIS DAN PERANCANGAN

Analisis adalah kegiatan penelitian yang dilakukan untuk menemukan kekurangan atau permasalahan yang ada dan kemudian penulis merumuskan dan memberikan solusi untuk penyelesaian masalah yang di dapat. Analisis dalam skripsi ini dilakukan dengan

mengobservasi sistem SMS gateway yang sedang berjalan dan mengumpulkan data log transaksi.

#### A. Analisis Masalah

Ketika ada permintaan SMS dari user yang dikirim oleh operator, SMS langsung ditangani oleh gateway in umumnya berupa file PHP yang akan menerima data SMS dari operator (protokolnya berupa HTTP dan SMPP). Selanjutnya, SMS akan dilempar ke server lain untuk diolah datanya, kemudian setelah diolah SMS akan dikirim lagi ke operator dengan menggunakan parameter-parameter yang sudah ditentukan. Dan terakhir data SMS di simpan ke basis data sebagai log transaksi. Aplikasi Gateway SMS ini berjalan secara *synchronous/langsung* (tidak terputus).

**Tabel 1.** Data log Transaksi periode Mei – Juni 2011

Bulan	Total Data SMS	Total service-id null
Mei	226,982	25,822
Juni	218,042	21,695

Data transaksi di atas adalah data yang diperoleh dari basis data transaksi. Dan untuk data lengkapnya bisa dilihat di lampiran. Berdasarkan tabel di atas, ada beberapa data yang *service-idnull* atau kosong karena sistem gagal melakukan *query* ke basis data pada saat proses SMS. Sehingga ketika SMS dikirim ke operator, akan gagal (invalid transaksi) karena data yang dikirim kurang lengkap. Semua itu menunjukkan bahwa sistem aplikasi SMS gateway yang sedang berjalan kurang optimal.

#### B. Alternatif Solusi

Dengan permasalahan yang sudah dikemukakan, maka ada beberapa solusi yang bisa dilakukan untuk mengatasinya:

1. Menambah komputer server yang digunakan sebagai pemroses SMS.
2. Menggunakan penyeimbang beban (*Load Balancing*), bisa berupa perangkat keras ataupun perangkat lunak.

#### 3. Membuat aplikasi antrian (Queue) SMS.

Dengan beberapa alternatif solusi yang diberikan, maka penulis memilih alternatif terakhir yaitu membuat aplikasi antrian. Karena dengan membuat aplikasi antrian, tidak akan ada biaya yang dikeluarkan oleh perusahaan. Dan semua permintaan SMS akan ditampung, sehingga tidak akan ada data SMS yang tidak valid ketika dikirim ke operator. Dan akan memudahkan dalam proses pemeliharaan aplikasi SMS Gateway itu sendiri. Dengan catatan bahwa proses SMS Gateway untuk operator Flexi berjalan seperti aplikasi yang sekarang berjalan.

#### C. Perancangan Sistem

Setelah dilakukan analisis terhadap sistem yang berjalan, solusi yang dipilih adalah membuat suatu aplikasi antrian. Dalam tahap perancangan inilah dijelaskan bagaimana sistem aplikasi antrian akan dibuat dengan spesifikasi proses-proses yang diterangkan secara rinci.

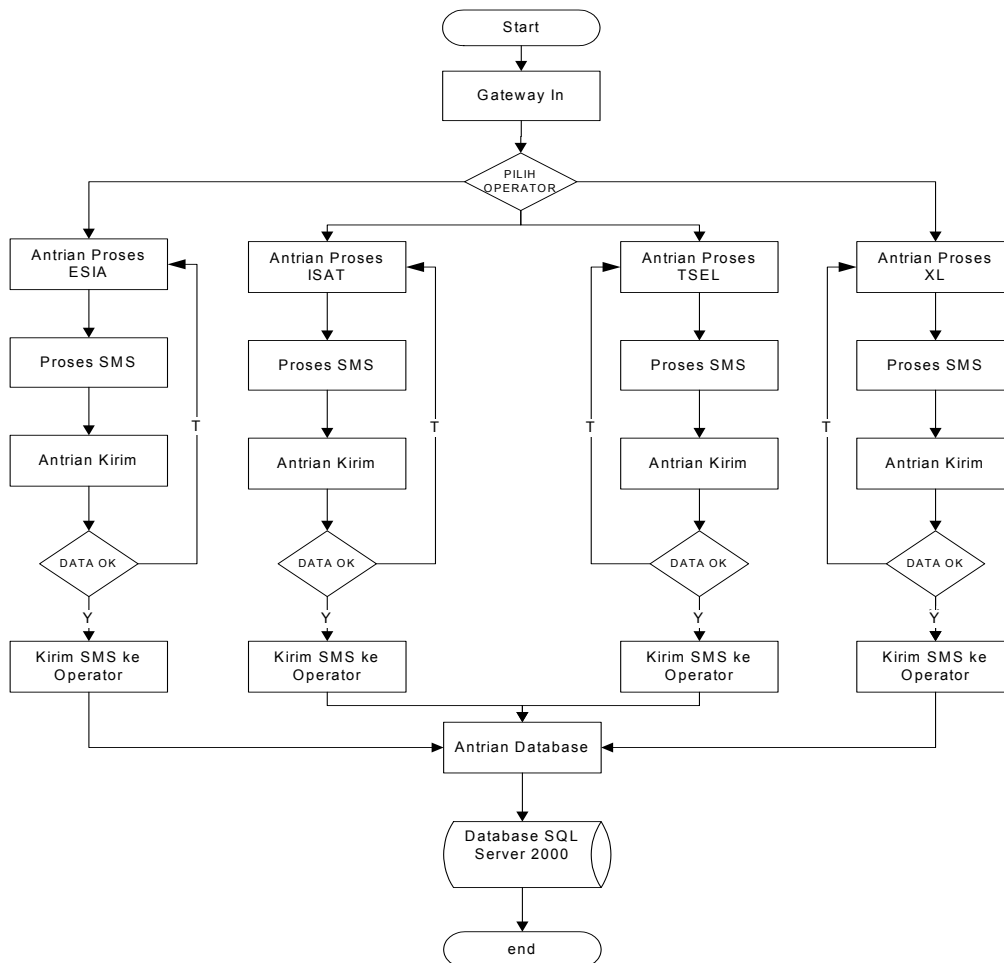
Metodologi penelitian yang digunakan untuk membuat aplikasi ini adalah *build and fix*. Dimana model ini sangat cocok untuk mengerjakan proyek yang berskala kecil. Dan aplikasi antrian ini, akan terus mengalami pengembangan sampai aplikasi antrian ini bisa menangani permintaan SMS yang sangat besar.

Sesuai dengan Gambar 2, maka konfigurasi aplikasi antrian yang digunakan yaitu *multiplechannel* dan *multi phase* sistem. Karena setiap pelanggan akan dilayani sesuai dengan operatornya (*multiplechannel*), ada 4 operator yaitu Esia, Indosat, Telkomsel dan XL. Dan setiap pelanggan akan mengalami 3 kali pemberhentian (*multi phase*) dari sistem antrian yaitu antrian proses, antrian kirim dan terakhir antrian basis data.

##### 1. Antrian Proses

Adalah antrian yang menampung data SMS sebelum data diproses. Datanya diperoleh dari Gateway In berupa parameter GET, yang diterima oleh aplikasi PHP. Kemudian data SMS tersebut dibuat menjadi file teks yang disimpan ke dalam suatu folder.





**Gambar 1.** Perancangan SMS gateway dengan menggunakan antrian *Multiple Channel* dan *MultiPhase* Sistem

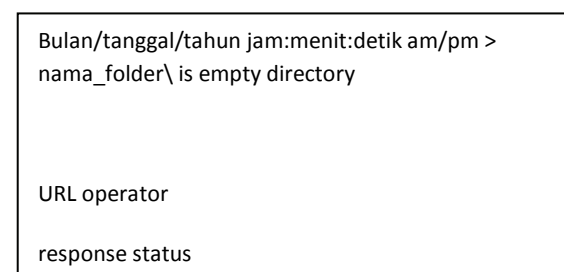
Aplikasi antrian proses akan berjalan di background untuk membaca file teks tersebut satu demi satu, mengirim datanya ke URL pemroses SMS kemudian menghapus file tersebut. Aplikasi ini akan membaca inputan file konfigurasi, yang berisi parameter-parameter yang digunakan dalam menjalankan program. Aplikasi ini berupa program konsol yang dijalankan melalui *command prompt* dan dibuat menggunakan bahasa pemrograman Delphi 7.

## 2. Antrian Kirim

Adalah antrian yang menampung data SMS setelah diproses. Sama seperti antrian proses, antrian kirim juga menggunakan file teks yang disimpan kedalam folder untuk menampung datanya. Data file teks dari antrian kirim, dibuat oleh pemroses SMS.

Aplikasi untuk membaca file teksnya sama seperti antrian proses menggunakan program konsol yang dijalankan melalui

*commandprompt*. Tugas utamanya adalah membaca file teks satu demi satu, mengirim datanya ke operator dan menghapus filenya apabila sudah dikirim. Untuk menyimpan alamat URL tiap operator beserta parameter yang dikirim, digunakanlah file konfigurasi.



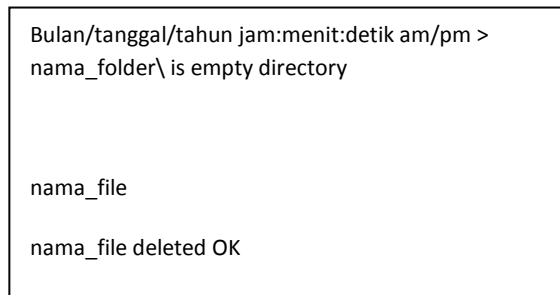
**Gambar 2.** Perancangan tampilan antrian kirim

## 3. Antrian Basis Data

Adalah antrian yang menampung data SMS untuk disimpan ke dalam basis data



transaksi. Sama seperti antrian proses dan kirim, antrian basis data juga menggunakan file teks yang disimpan dalam folder. File teksnya dibuat oleh aplikasi antrian kirim. Antrian basis data akan membaca file teks dan menyimpannya kedalam basis data transaksi. Koneksi ke basis data *SQL Server2000* menggunakan komponen ADO (*ActiveXObject*).



**Gambar 3**Perancangan tampilan antrian basis data

#### 4. IMPLEMENTASI DAN PENGUJIAN

##### A. Implementasi

##### 1. Potongan Coding Program Antrian Kirim

###### a) Program PHP

Program PHP ini digunakan untuk membuat file teks SMS antrian proses. Datanya diperoleh dari Gateway In berupa parameter (msisdn, sms, trx\_id, trx\_date, operator, sdc, channel, ip) yang dikirim dengan metode GET. File teks disimpan dalam suatu folder dan dipisahkan berdasarkan nama operator. Satu data SMS akan menjadi satu teks file.

Baris pertama potongan coding dibawah adalah membaca parameter yang dikirim dari *Gateway In* dengan menggunakan metode GET pada *httprequest*. Baris selanjutnya adalah membuat variabel yang digunakan sebagai tempat penyimpanan file teks/folder. Terakhir adalah proses pembuatan file teks menggunakan *class PushDocument*. Perintah *unset* digunakan untuk *destroy objectclass* yang sudah dibuat. Dan *exit* untuk keluar dari program dengan menampilkan pesan "finish" dilayar.

```
<?php
$msisdn =
!empty($_GET['msisdn']) ? $_GET['msisdn'] : "";
if ($operator == "ISAT")
    $outboxdir =
$home_dir."/inbox/QueueProccess/ISAT";
```

```
elseif ($operator == "TSEL")
    $outboxdir =
$home_dir."/inbox/QueueProccess/TSEL";
$doc = new PushDocument(160);
$doc->setDataSMSIN($msisdn, $sms, $trx_id,
$trx_date, $operator, $sdc, $channel,
$meta_data, $ip_tsel);
$doc->store($outboxdir);
unset($doc);
exit("finish");
?>
```

###### b) Program Delphi

Program pascal/delphi ini digunakan untuk membaca file teks dalam folder antrian proses, mengirim ke URL pemroses dan menghapusnya. Variabel yang sifatnya dapat berubah, disimpan dalam suatu file konfigurasi yang dibaca ketika program dijalankan.

Variabel *i=0* digunakan pada pengulangan *repeat until*, agar program membaca folder antrian proses secara berulang-ulang. Potongan coding if pertama digunakan untuk membaca file teks pada folder sampai folder tersebut kosong dikombinasikan dengan pengulangan *repeat until*.

*Pushdata* adalah file ini yang dibuat dari file teks dalam antrian. Sdc adalah variabel yang dibaca dari file ini *pushdata* pada tag [SMSQUEUE] bagian sdc. Yurl adalah alamat URL pemroses SMS yang dibaca dari file konfigurasi ketika aplikasi dijalankan. Xurl diisi dari variabel yurl, kemudian dilakukan *replace/penggantian* isi variabel dalam format @@ dengan variabel yang dibaca dari file *pushdata*. Jika variabel *msisdn* pada file *pushdata* tidak kosong dan variabel *live* pada file konfigurasi diisi yes, maka program akan mengirim data ke pemroses SMS dan menampilkan response yang diterima pada layar. Yang paling terakhir adalah menghapus file teks dalam antrian. Jika berhasil maka layar akan menampilkan pesan "namafile deleted OK" dan selain itu akan menampilkan pesan "tanggal/bulan/tahun jam folder\_antrian is empty directory". Serta diberi jeda waktu untuk melakukan pembacaan lagi selama 3 detik.

```
i := 0;
repeat
begin
if FindFirst(smsin+'*.*', faAnyFile,
searchResult) = 0 then
begin
repeat
```



```

        namafile := searchResult.Name ;
        until FindNext(searchResult) <> 0;
        FindClose(searchResult);
    end;
    pushdata := TIniFile.Create(smsin+namafile)
    ;
    sdc :=
    pushdata.ReadString('SMSQUEUE','sdc,');
    yurl := appINI.ReadString('SDC','url_'+sdc, '');
    xurl := yurl;
    xurl:=StringReplace(xurl,'@@msisdn@@',pus
    hdata.ReadString('SMSQUEUE','msisdn',''),[rfR
    eplaceAll,rflgnoreCase]);
    Writeln(xurl);
    if pushdata.ReadString('SMSQUEUE','msisdn',
    '') <> '' then
    begin
        if live = 'yes' then
            opr_ret := GetUrl(xurl);
            Writeln(opr_ret);
        end;
        Writeln;
        if DeleteFile(smsin+namafile) then
            Writeln(smsin+namafile+' deleted OK')
        else
            begin
                Writeln( DateTimeToStr(now)+'>' + smsin+'
                is empty directory. ');
                Sleep( 3000 );
            end;
        end;
    until i > 0;

```

#### c) Format file teks

Format file teks yang digunakan pada antrian proses adalah sebagai berikut:

```

[SMSQUEUE]
msisdn=6285222022667
sms=REG AMEL
trx_id=0sms11131286076061248340209
trx_date=08-09-2011 8:29:42 AM
operator=TSEL
sdc=9886
channel=SMS
meta_data=

```

## 2. Potongan Coding Program Antrian Kirim

#### a) Program PHP

Program PHP ini akan disisipkan pada pemroses SMS, dan digunakan untuk membuat file teks SMS antrian kirim. Data file teks dipisahkan berdasarkan nama operator.

Kondisi if digunakan untuk membedakan folder penyimpanan antrian kirim berdasarkan nama operatornya. File teks dibuat dari

*ObjectClass* PullDocument, dan disimpan pada folder antrian yang sudah didefinisikan sebelumnya. Selanjutnya, koneksi ke basis data diputus, kemudian *Object* dari *Class* *SQLServer* dan *PullDocument* dihapus dan keluar dari program pemroses SMS.

```

<?php
if ($operator == "TSEL")
    $outboxdir =
    $home_dir."/inbox/QueueSend/TSEL";
if ($operator == "ISAT")
    $outboxdir =
    $home_dir."/inbox/QueueSend/ISAT";

$doc = new PullDocument(160);
$doc-
>setDataSMSPULL($ip_tsel,$keyword,$sms,$c
hannel,$operator,$sdc,$msisdn,$sid,$reply,
$trx_id,$tarif,$keyword_status,$program_id,$xl
_short_name=",$xl_content_type=");
$doc->store($outboxdir);

$sqlserver->disconnect();
unset($sqlserver);
unset($doc);
flush();
exit();
?>

```

#### b) Program Delphi

Secara garis besar antrian kirim ini hampir sama dengan antrian proses. Program ini digunakan untuk membaca file teks dalam folder. Sebelum dikirim, data dicek terlebih dahulu. Jika tag sid kosong, maka data akan dikembalikan ke antrian proses dan menampilkan pesan 'movedtoQueueOK'. Sedangkan jika lengkap, data dikirim ke operator dan menampilkan pesan response dari operator kemudian membuat file teks antrian basis data dan terakhir menghapus data file teks pada antrian kirim dengan menampilkan pesan dilayar "namafile deletedOK". Pada kondisi folder kosong, diberi jeda waktu untuk pembacaan lagi selama 3 detik.

Ada perlakuan khusus untuk operator Telkomsel. URL yang digunakan ada 2 yaitu untuk *PullRequest* (SMS yang dikirim dari permintaan pelanggan) dan *PushRequest* (SMS yang dikirim pada pelanggan tanpa mengirim SMS terlebih dahulu karena sudah terdaftar pada suatu layanan). Pada operator lain, semua permintaan pelanggan akan menggunakan mekanisme *pull*. Sedangkan operator Telkomsel untuk permintaan pelanggan yang melakukan registrasi



menggunakan mekanisme push. Sehingga pada file konfigurasi dibuat 2 URL untuk operator Telkomsel yaitu untuk *pull* dan *push*.

```
i:=0;
repeat
begin
if
FindFirst(smsbox+'*.*',faAnyFile,searchResult)
=0 then
begin
repeat
namafile:=searchResult.Name;
untilFindNext(searchResult)<>0;
FindClose(searchResult);
end;
pushdata:=TIniFile.Create(smsbox+namafile)
;
opr:=pushdata.ReadString('SMSOUT','operator',
'');
if(pushdata.ReadString('SMSOUT','operator','')
='TSEL')and(pushdata.ReadString('SMSOUT','t
ype','')='PULL') then

yurl:=appINI.ReadString('SDC','url_'+opr+'_p
ull','')
else
yurl:=appINI.ReadString('SDC','url_'+opr,'');
xurl:=yurl;
xurl:=StringReplace(xurl,'@@ip@@',pushdata
.ReadString('SMSOUT','ip',''),[rfReplaceAll,rflgn
oreCase])
if (namafile<>'..') then
begin
if pushdata.ReadString('SMSOUT','sid','')=""
then
begin
smsfile:=TIniFile.Create(qbox+namafile);
with smsfile do
begin

WriteString('SMSQUEUE','msisdn',pushdata.Re
adString('SMSOUT','msisdn',''));
end;
if DeleteFile(smsbox+namafile) then
Writeln(smsbox+namafile+'movedtoQueueOK
');
end;
end;
if pushdata.ReadString('SMSOUT','sid','')<>"
then
begin
if live='yes' then opr_ret:=getUrl(xurl);
Writeln(opr_ret);
dlrfile:=TIniFile.Create(dlrbox+namafile);
with dlrfile do
begin
```

```
WriteString('SMSADO','keyword',pushdata.Rea
dString('SMSOUT','keyword',''));
end;
end;
if DeleteFile(smsbox+namafile) then
Writeln(smsbox+namafile+'deletedOK')
else
begin

Writeln(DateTimeToStr(now)+'>'+smsbox+'is
emptydirectory. '); Sleep(3000);
end;
end;
until i>0;
```

### c) Format file teks

Format file teks yang digunakan pada antrian kirim adalah sebagai berikut:

#### [SMSOUT]

```
ip=202.3.219.3
keyword=UNREG
fullsms=UNREG
channel=SMS
operator=TSEL
sdc=9886
msisdn=6282157432862
sid=MDS_Y_C04000_1100_PULL
text=Selamat siang 6282157432862,
```

Perintah tidak dikenali. Info lebih lanjut hubungi CS:0217201478

```
tid=0sms11131286132622648343588
tarif=0
type=PULL
programid=-1
short_name=
content_type=
```

#### [ERROR]

```
desc=
maxretry=10
currretry=
lastretry=
```

### 3. Potongan Coding Program Antrian Basis Data

#### a) Program Delphi

Program pascal/delphi ini digunakan untuk membaca file teks dalam folder, memasukkan kedalam basis data dan menghapus file teksnya. Koneksi ke basis data sendiri menggunakan komponen ADO (*ActiveXObject*). Sama seperti antrian proses dan antrian kirim, program antrian basis data dijalankan pada *commandprompt* dan menggunakan file konfigurasi yang sama dengan file konfigurasi yang digunakan untuk antrian kirim.

Program berjalan secara berulang menggunakan perulangan repeat until. Awalnya dilakukan pencarian ke folder antrian basis data. Jika folder tidak kosong, variabel namafile akan diisi. Variabel dlrfile dibuat, dengan membaca isi file teks yang ada. Selanjutnya dilakukan mekanisme try membuat perintah *querysql* untuk insert ke tabel transaksi yaitu sms\_in, sms\_out dan sms\_dlr.

Bagian paling akhir, jika file teks dapat dihapus maka menampilkan pesan 'deletedOK'. Selain itu layar akan menampilkan pesan 'is empty directory.' dan koneksi ke basis data diputus. Untuk pembacaan ulang folder diberi jeda 8 detik dan koneksi ke basis data dibuka kembali.

```
i:=0;
repeat
begin
if
FindFirst(dlrbox+'*.*',faAnyFile,searchResult)
=0 then
begin
repeat
namafile:=searchResult.Name;
untilFindNext(searchResult)<>0;
FindClose(searchResult);
end;
dlrfile:=TIniFile.Create(dlrbox+namafile);
Writeln(namafile);
Try with query1 do
begin
sqlText:='INSERT INTO
sms_in(in_id,keyword,msisdn,bnumber,fullsms
,operator_id,dtm,trx_date,binary,channel)'+
'VALUES(:in_id,:keyword,:msisdn,:bnumber,:ful
lsms,:operator_id,:dtm,:trx_date,:binary,:chann
el)';
CommandText:=sqlText;
CommandType:=cmdText;
Parameters.ParamByName('in_id').Value:=dlrfi
le.ReadString('SMSADO','tid','');
Execute;
sqlText:='INSERT INTO
sms_out(out_id,in_id,keyword,trans_id,msisdn,
reply,reply_status,reply_type,sender,dtm,trx_d
ate,program_id,service_id,submit_dtm,operato
r_id,cpremark)'+
'VALUES(:out_id,:in_id,:keyword,:trans_id,:msis
dn,:reply,:reply_status,:reply_type,:sender,:dtm
,:trx_date,:program_id,:service_id,:submit_dtm,
:operator_id,:cpremark)';
CommandText:=sqlText;
CommandType:=cmdText;
```

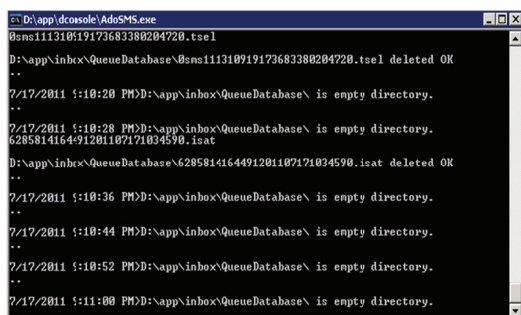
```
Parameters.ParamByName('out_id').Value:=dlr
file.ReadString('SMSADO','tid','');
If (dlrfile.ReadString('SMSADO','type','')='REG')
or
(dlrfile.ReadString('SMSADO','type','')='UNREG
')
or
(dlrfile.ReadString('SMSADO','type','')='PULL')
then
replytype:='PULL'
else
replytype:='PUSH';
Execute;
sqlText:='INSERT INTO
sms_dlr(trans_id,msisdn,service_id,dlr_status,
dlr_dtm,smcid,msg_status,operator_id)'+
'VALUES(:trans_id,:msisdn,:service_id,:dlr_stat
us,:dlr_dtm,:smcid,:msg_status,:operator_id)';
CommandText:=sqlText;
CommandType:=cmdText;
xdlr_status:=dlrfile.ReadString('SMSADO','dlr_
status','');
if
(dlrfile.ReadString('SMSADO','operator','')='TS
EL') and (LeftStr(xdlr_status,1)='1') then
xdlr_status:='1';
if
dlrfile.ReadString('SMSADO','operator','')='ISA
T' then
xdlr_status:='-';
if
xdlr_status:=dlrfile.ReadString('SMSADO','ok_s
tatus','') then
msg_status:='DELIVRD'
else
msg_status:=xdlr_status;
Parameters.ParamByName('operator_id').Valu
e:=dlrfile.ReadString('SMSADO','operator','');
Execute;
end;
except
end;
Writeln;
if DeleteFile(dlrbox+namafile) then
Writeln(dlrbox+namafile+'deletedOK')
else
begin
Writeln(DateTimeToStr(now)+'>'+dlrbox+'is
empty directory. ');
conn.Close; Sleep(8000);
conn.Open; query1.Connection:=conn;
end;
end;
until i>0;
```



b) Format file teks

Format file teks yang digunakan pada antrian basis data adalah sebagai berikut:

```
[SMSADO]
keyword=UNREG.CG
fullsms=UNREG.CG
channel=SMS
operator=TSEL
sdc=9886
ok_status=1
msisdn=6285379574921
sid=MDS_Y_C04100_1100_PUSH
text=Selamat siang 6285379574921, Perintah
tidak dikenali. Info lebih lanjut hubungi
CS:0217201478
tid=1sms11131286453899364816376
tarif=0
type=UNREG
programid=-1
dlr_status=3:6:3:21
dtm=08-09-2011
trx_date=08-09-2011 10:37:39 AM
```



Gambar 4 Aplikasi antrian basis data

## B. Pengujian

Menjelaskan lingkungan pengujian yaitu untuk menguji perangkat lunak yang dihasilkan. Sebelum pengujian terhadap aplikasi ini dilakukan, terlebih dahulu harus dilakukan perencanaan untuk melakukan pengujian tersebut sehingga pengujian yang akan dilakukan akan sesuai dengan ketentuan yang ada, jadi tidak terjadi penyimpangan dari tujuan yang telah dibuat sebelumnya. Pengujian yang dilakukan menggunakan metode *black-box*.

### 1. Lingkungan Pengujian

Perangkat keras yang digunakan dalam pengujian adalah:

- a) Server Supermicro X8SIL
- b) Processor Intel Xeon X3430 @2,40Ghz.
- c) Memory 8 GB.

d) Harddisk 250 GB.

Perangkat lunak yang digunakan dalam pengujian adalah:

- a) Sistem Operasi Windows Server 2003 R2 Enterprise Edition.
- b) Basis data Microsoft SQL Server 2000.
- c) Webserver Apache.
- d) Bahasa pemrograman PHP dan Pascal/Delphi 7.

## 2. Pengujian

Pengujian dilakukan menggunakan blackbox yaitu pengujian yang dilakukan dari sisi pengguna untuk memastikan bahwa sistem bekerja sebagaimana mestinya sesuai dengan yang diharapkan tanpa perlu mengetahui secara detail bagaimana sistem tersebut dibuat.

Antrian proses diuji hanya dengan pengujian yang bersifat normal karena datanya berasal dari operator sehingga kemungkinan kecil jika data yang didapat tidak normal. Begitu pula untuk antrian basis data diuji dengan data normal, karena apabila data tidak lengkap antrian kirim akan mengembalikannya ke antrian proses. Sedangkan untuk antrian kirim, akan diuji dengan normal dan tidak normal. Dengan pengujian tidak normal pada antrian kirim, data SMS yang tidak lengkap dapat kembali ke antrian proses untuk dikirim ulang ke pemroses SMS sampai data lengkap/valid.

Masing-masing item yang diuji akan diulang sebanyak 4 kali sesuai dengan operator yang ada (Esia, ISAT, TSEL dan XL), dimana mewakili dari sistem antrian *multiplechannel* (ada 4 pelayan berdasarkan operator) dan *multiphase* sistem (ada 3 pemberhentian: antrian proses, antrian kirim dan antrian basis data).

**Tabel 2.** Hasil pengujian operator Esia

No	Item yang Diuji	Sifat Pengujian	Cara Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Antrian proses	Normal	Membuat file teks pada folder antrian proses Esia.	Data dikirim ke URL pemroses Esia untuk diolah dan hasilnya akan masuk kedalam folder antrian kirim Esia.	Sesuai
2.	Antrian kirim	Normal	Membuat file teks pada folder antrian kirim Esia, dengan data yang lengkap.	Data dikirim ke URL operator Esia dan mendapat response dan disimpan kedalam folder antrian basis data.	Sesuai
3.	Antrian kirim	Tidak normal	Membuat file teks pada folder antrian kirim Esia, dengan data tidak lengkap (service id kosong).	Data dikembalikan ke folder antrian proses Esia.	Sesuai
4.	Antrian basis data	Normal	Membuat file teks pada folder antrian basis data.	Data masuk kedalam basis data transaksi, yaitu disimpan ketabel sms_in, sms_out, sms_dlr.	Sesuai

**Tabel 3.** Hasil pengujian operator Indosat

No	Item yang Diuji	Sifat Pengujian	Cara Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Antrian proses	Normal	Membuat file teks pada folder antrian proses Indosat.	Data dikirim ke URL pemroses Indosat untuk diolah dan hasilnya akan masuk kedalam folder antrian kirim.	Sesuai
2.	Antrian kirim	Normal	Membuat file teks pada folder antrian kirim Indosat, dengan data yang lengkap.	Data dikirim ke URL operator Indosat dan mendapat response dan disimpan kedalam folder antrian basis data.	Sesuai
3.	Antrian kirim	Tidak normal	Membuat file teks pada folder antrian kirim Indosat, dengan data tidak lengkap (service id kosong).	Data dikembalikan ke folder antrian proses Indosat.	Sesuai
4.	Antrian basis data	Normal	Membuat file teks pada folder antrian basis data.	Data masuk kedalam basis data transaksi, yaitu disimpan ketabel sms_in, sms_out, sms_dlr.	Sesuai



**Tabel 4.** Hasil pengujian operator Telkomsel

No	Item yang Diuji	Sifat Pengujian	Cara Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Antrian proses	Normal	Membuat file teks pada folder antrian proses Telkomsel.	Data dikirim ke URL pemroses Telkomsel untuk diolah dan hasilnya akan masuk kedalam folder antrian kirim.	Sesuai
2.	Antrian kirim	Normal	Membuat file teks pada folder antrian kirim Telkomsel, dengan data yang lengkap.	Data dikirim ke URL operator Telkomsel dan mendapat response dan disimpan kedalam folder antrian basis data.	Sesuai
3.	Antrian kirim	Tidak normal	Membuat file teks pada folder antrian kirim Telkomsel, dengan data tidak lengkap (service id kosong).	Data dikembalikan ke folder antrian proses Telkomsel.	Sesuai
4.	Antrian basis data	Normal	Membuat file teks pada folder antrian basis data.	Data masuk kedalam basis data transaksi, yaitu disimpan ketabel sms_in, sms_out, sms_dlr.	Sesuai
5.	Pengujian pembebanan	Normal	Program masuk ke cek pulsa Telkomsel (*123#).	Data SMS yang hilang dapat dikurangi.	Sesuai, data SMS yang hilang persentasenya kecil (1%).

**Tabel 5.** Hasil pengujian operator XL

No	Item yang Diuji	Sifat Pengujian	Cara Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Antrian proses	Normal	Membuat file teks pada folder antrian proses XL.	Data dikirim ke URL pemroses XL untuk diolah dan hasilnya akan masuk kedalam folder antrian kirim.	Sesuai
2.	Antrian kirim	Normal	Membuat file teks pada folder antrian kirim XL, dengan data yang lengkap.	Data dikirim ke URL operator XL dan mendapat response dan disimpan kedalam folder antrian basis data.	Sesuai
3.	Antrian kirim	Tidak normal	Membuat file teks pada folder antrian kirim XL, dengan data tidak lengkap (service id kosong).	Data dikembalikan ke folder antrian proses XL.	Sesuai
4.	Antrian basis data	Normal	Membuat file teks pada folder antrian basis data.	Data masuk kedalam basis data transaksi, yaitu disimpan ketabel sms_in, sms_out, sms_dlr.	Sesuai

### 3. Analisis Hasil Pengujian

Pembagian antrian berdasarkan *channel* operator, akan bermanfaat pada saat proses

perbaikan berlangsung. Jika ada salah satu operator yang mengalami gangguan teknis, maka aplikasi SMS dapat diberhentikan

dengan menghentikan program antrian operator yang sedang mengalami gangguan teknis tanpa harus memberhentikan semua layanan SMS yang sedang berjalan.

Sedangkan pembagian antrian berdasarkan phase/pemberhentian, akan bermanfaat pada saat terjadi perbaikan diaplikasi internal sistem. Misalnya proses backup basis data transaksi sedang berlangsung, maka hanya dengan menghentikan aplikasi antrian basis data, aplikasi SMS gateway secara keseluruhan dapat terus berjalan. Karena data SMS akan ditampung di folder antrian basis data.

## 5. KESIMPULAN DAN SARAN

### A. Kesimpulan

Dengan menggunakan aplikasi antrian, data SMS yang hilang dapat dikurangi. Berdasarkan hasil pengujian, persentase *service-id* yang null (kosong) selama bulan Juli 2011 sebesar 0,32%. Sangat jauh berkurang dibanding periode transaksi bulan sebelumnya, yaitu pada bulan Mei 2011 sebesar 11,38% dan bulan Juni 2011 sebesar 9,95%.

### B. Saran

Aplikasi antrian dapat mengurangi data SMS yang hilang. Akan tetapi apabila permintaan terhadap layanan SMS sangat besar pada saat yang bersamaan, akan terjadi *delay* waktu yang lama bagi

pelanggan untuk mendapatkan balasan SMS. Sehingga kualitas pelayanan terhadap pelanggan berkurang. Apabila memungkinkan dari segi biaya, masalah diatas bisa diatasi dengan menerapkan penyeimbang beban (*LoadBalancing*). Misalnya dengan menambahkan mesin pemroses SMS. Yaitu membagi kerja pemroses SMS dengan minimal 2 server yang akan mempercepat proses SMS sehingga *delay* waktu balasan SMS ke pelanggan lebih cepat.

## DAFTAR PUSTAKA

- [1] Haverkort, Boudewijn R. *Performance of Computer Communication System*. German: John Wiley & Sons Ltd, 1998.
- [2] Imron Rozidi, Romzi. *Membuat Sendiri SMS Gateway Berbasis Protokol SMPP*. Yogyakarta: ANDI, 2004.
- [3] Keretho, Somnuk. *Introduction to Software Engineering*. Retrieved 31 July 2011, <http://www.cpe.ku.ac.th/~sk/204541/intro-to-software-engineering.PDF>.
- [4] Ng Chee-Hock and Soong Boon-Hee. *Queueing Modelling Fundamentals With Applications in Communication Network*. Hongkong: John Wiley & Sons Ltd, 2008.
- [5] Wahyujati, Ajie. *Riset Operasional 2 – Model Antrian*. Retrieved 30 Juni 2011, <http://ajiew.staff.gunadarma.ac.id/Downloads/files/8624/Model+Antrian.pdf>.