

APLIKASI MOBILE SMS GATEWAY OVER GPRS UNTUK MENGAkses SISTEM INFORMASI

Wahyu Pujiyono¹, Arif Rahman², Ananta Wijaya¹

¹Program Studi Teknik Informatika, Universitas Ahmad Dahlan,

²Program Studi Sistem Informasi Universitas Ahmad Dahlan,

Kampus III UAD Jl. Prof. Dr. Soepomo, Janturan, Umbulharjo, Yogyakarta

Telp 0274-379418, 381523, Fax. 0274- 381523

email: yywahyup@yahoo.com, arif@uad.ac.id, nant3xad@yahoo.com

Abstract

SMS via GPRS technology can be used to access data from information systems. This technology is one of regular and premium SMS extensions. The SMS over GPRS is composed by three components: the mobile application, server, router and gateway of information systems. The research objective was to build a mobile application to access information systems with the assumption that the server and the gateway router of information systems already exist. System was built using the http protocol, which has a request-response mechanism. There are two methods in the delivery of the message (request) and the receipt of the message (response) from the server, the message filtering and message parsing. Filtering is used to filter the message text /string that not known to be a string known by http, and then in message parsing process, message parses in the form of XML.

Keywords: GPRS, SMS, http, request-response, message filtering, message parsing

Abstrak

SMS menggunakan fasilitas GPRS untuk mengakses sistem informasi merupakan salah satu pengembangan dari SMS regular dan premium, pada dasarnya SMS GPRS terbagi menjadi tiga komponen yaitu aplikasi mobile, server router dan gateway sistem informasi. Tujuan dari penelitian ini membangun aplikasi mobile untuk mengakses sistem informasi dengan asumsi server router dan gateway sistem informasi sudah ada. Sistem yang dibangun menggunakan protokol http, yang memiliki mekanisme request-response. Terdapat dua metode dalam pengiriman pesan (request) dan penerimaan pesan (response) dari server, yaitu filtering pesan dan parsing pesan. Filtering pesan digunakan untuk memfilter teks/string yang tidak dikenali http menjadi string yang dikenali http kemudian pesan yang diterima dalam bentuk xml akan di parsing sesuai pesan dalam tag xml yang ingin ditampilkan.

Kata kunci: GPRS, SMS, http, request-response, filtering pesan, parsing pesan

1. PENDAHULUAN

Aplikasi berbasis SMS dewasa ini telah populer digunakan dalam dunia aplikasi mobile. Aplikasi ini menggunakan protokol dan gateway tersendiri. Beberapa tahun belakangan ini seiring dengan perkembangan teknologi GPRS, telah dikembangkan pula pengiriman SMS melalui jalur GPRS. Bahkan peralatan rumah pun bisa dikendalikan menggunakan gateway rumah melalui modem GPRS menggunakan SMS [1].

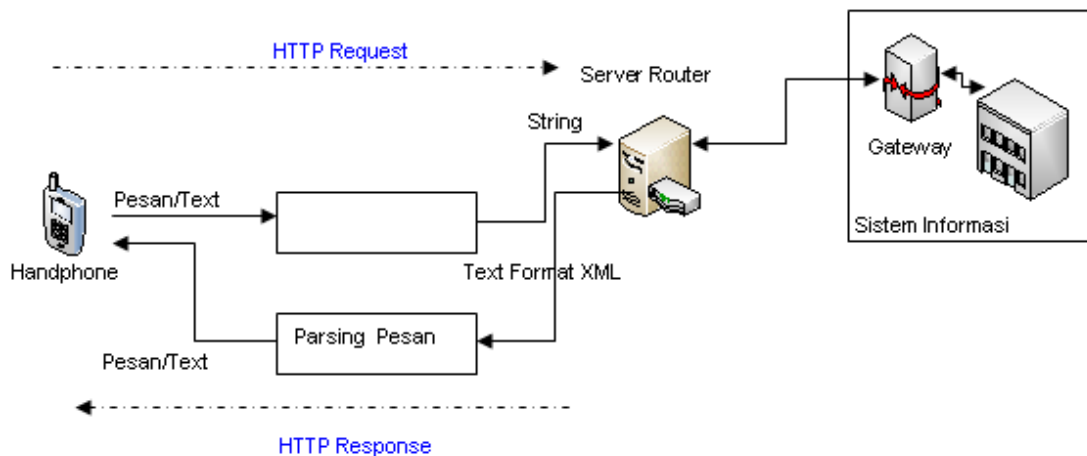
Dalam aplikasi SMS yang menggunakan konektivitas GPRS, operator memberi jalan bagi *mobile-phone* dan *Web Server* untuk saling bertukar data. Selain memberi jalur pertukaran data pihak operator juga berperan dalam pemberian alamat IP (*Internet Protocol*) kepada setiap *mobile-phone* agar dapat berkomunikasi dengan jaringan *internet*.

Aplikasi SMS menggunakan media konektivitas GPRS telah ada sebelumnya yaitu digunakan untuk aplikasi *chatting* [2]. Pada penelitian ini kami mengembangkan aplikasi *chatting* untuk mengakses sistem informasi pada suatu instansi/perusahaan, yaitu dengan mengirimkan pesan sesuai format pesan yang telah ditentukan melalui *mobile-phone* sebagai klien, kemudian *request* klien akan dikirimkan ke *gateway* sistem informasi melalui *server router*, dan balasan dari *gateway* sistem informasi akan dikembalikan ke *server router* kemudian

dikirimkan ke klien dan ditampilkan pada layar *handphone*, sesuai dengan *request* yang diminta [3].

2. METODE PENELITIAN

Penelitian dilakukan dengan cara membangun aplikasi *mobile* untuk SMS *gateway over GPRS*. Aplikasi tersebut dalam komunikasi datanya menggunakan protokol *http* dengan mekanisme *request-response*. Pesan yang dikirim user melalui perangkat *mobile* oleh aplikasi ini akan difilter untuk mendapatkan format data yang kompatibel dengan standar *http*. Kemudian pesan yang telah ter-filter dikirimkan ke *webserver* melalui koneksi *GPRS* untuk diproses lebih lanjut. *Response* dari *webserver* berupa data hasil pemrosesan, akan dikirimkan ke *mobile client* dalam bentuk *XML*. *Response* tersebut oleh aplikasi ini akan diparsing berdasarkan *tag-tag* yang bersesuaian sehingga didapatkan teks pesan. Cara kerja sistem ini ditunjukkan oleh Gambar 1. Aplikasi *mobile* dikembangkan dengan Teknologi *Java 2 ME*, *CLDC 1.0*, *MIDP 1.0/2.0*. [4] Pembuatan aplikasi dilakukan dengan *PC* dengan lingkungan pengembangan *Windows*, *Java Development Kit 6.0* dan *IDE Java Mobile Edition* dalam *Netbeans 6.5*. [5].



Gambar 1. Blok diagram sistem SMS Over GPRS

Algoritma Filter Pesan { Diberikan inputan n }
Deklarasi x[] : Char n : String {Input}
Deskripsi Read(x[]) for i <- 0 to n.lenght do Case (x[i]) ' ' : karakter <- '%20' '#' : karakter <- '%23' ' " ' : karakter <- '"' '&' : karakter <- '&' '<' : karakter <- '<' '>' : karakter <- '>' ' ' ' : karakter <- '&apos' Endcase Return(x[])

Gambar 2. Algoritma pem-filter-an pesan

2.1. Filtering Pesan

Proses pengiriman pesan (*request*) pada sms over gprs menggunakan http, akan melalui tahap filtering pesan yang berfungsi untuk mengenali teks/string pesan agar bisa dibaca oleh standar http, sehingga apabila ditemukan teks/string yang tidak dikenali maka program akan menjalankan filterisasi atau merubah teks/string tersebut menjadi format http. Untuk lebih jelasnya bisa ditunjukkan pada Tabel 1. Dari ketentuan dalam Tabel 1, dibuatlah sebuah fungsi menggunakan bahasa pemrograman Java untuk melakukan filter pesan sesuai Gambar 2.

Tabel 1. Filtering Pesan

Teks/String	Format HTTP
<spasi>	%20
#	%23
" (tanda petik ganda)	"
&	&
< , >	< , >
` (tanda petik tunggal)	'

2.2 Parsing Pesan

Pesan yang diterima dari server (*response*) dalam format XML, maka dibutuhkan fungsi untuk memarsing format XML tersebut sehingga diperoleh isi pesannya. Format XML yang diterima terbagi menjadi 3 yaitu no tujuan, isi pesan, dan jenis request, yang ketiganya dipisahkan berdasarkan tag-tag. Parsing XML diterapkan pada pemrograman java, dalam hal ini peneliti menggunakan kXML 1.0 sebagai library tambahan. Algoritma fungsi parsing XML yang digunakan adalah seperti ditunjukkan pada Gambar 3. Dari algoritma itu jelas bahwa pesan yang akan ditampilkan hanya teks yang terdapat dalam tag body.

Algoritma Parsing Pesan { Diberikan inputan x }
Deklarasi x : String {input} i : false {boolean} t : String {xml type}
Deskripsi Read(x) Case (t) t : Start_Tag <- if 'body' then i <- true t : End_Tag <- if 'body' then i <- false t : Text <- if i then 'x' Endcase

Gambar 3. Algoritma fungsi parsing XML

Sistem SMS gateway over GPRS untuk mengakses sistem informasi mempunyai 3 komponen perangkat lunak yang harus terpenuhi yaitu:

a. Aplikasi sms pada *mobile-phone* sebagai klien

Aplikasi SMS ini merupakan aplikasi *java* dengan nama filenya *sigawe.jar* yang harus diinstal ke *mobile-phone*, dan dapat mengirimkan *request* ke sever untuk meminta layanan sistem informasi, kemudian menerima balasan dari *server* dan ditampilkan pada layar. Persyaratan minimum *mobile-phone* yang harus dipenuhi, yaitu: *mobile-phone* sudah mendukung Java CLDC 1.0 dan MIDP 1.0/2.0, terdapat konektivitas GPRS yang sudah tersetting pada *mobile-phone* maupun pada kartu operator yang digunakan dan *mobile-phone* bisa digunakan untuk akses *internet*.

b. Server

Server merupakan aplikasi yang bertugas untuk mengatur lalu lintas pertukaran data antara klien *mobile* dengan *gateway*, sehingga harus selalu terkoneksi *internet*, dan mempunyai IP

public yang bisa diakses oleh klien *mobile* dan *gateway* sistem informasi. Pada sistem SMS over GPRS ini, *server* yang digunakan pada penelitian ini adalah *server socket sigawe*.

c. *Gateway*

Gateway yang digunakan dalam penelitian ini adalah *gateway socket sigawe*. Untuk *gateway* sistem informasi swalayan, data yang diterima *gateway* merupakan *request* dari klien *mobile* dalam meminta informasi, sehingga pesan yang dikirimkan klien melalui *server* akan diterima *gateway* dan kemudian diproses untuk menghasilkan *output* sesuai dengan *request* yang diminta, dan dikembalikan lagi ke *server* untuk disampaikan ke klien *mobile*.

3.1 Proses Pengiriman Pesan

Untuk pengiriman pesan menggunakan HTTP GET, sehingga sistem yang digunakan secara *request-response*, yaitu pengguna mendapatkan informasi dari sistem informasi setelah pengguna melakukan *request*, kemudian *server* memberikan *response* sesuai dengan permintaan pengguna, artinya *server* tidak bisa memberikan informasi atau melakukan *broadcast* apabila tidak ada *request* dari pengguna, kemudian koneksi terputus.

3.1.1 Membuat Thread Kirim Pesan & Penerimaan Pesan

Diperlukan thread untuk penanganan pengiriman pesan sehingga proses pengiriman pesan bisa berjalan secara *asynchronous*, ini dimaksud bahwa setiap mengirim dan menerima pesan akan membuat thread baru. Gambar 4 menunjukkan *thread* untuk pengiriman pesan dan menerima pesan.

Pada Gambar 4, menjelaskan bahwa pesan sebelum dikirim akan difilter menggunakan fungsi *FilterPesan* pada Gambar 5 yang kemudian ditampung pada variabel *pes* yaitu pada baris 10. Proses mengirim pesan dan menerima pesan pada thread diharuskan menggunakan fungsi *try-catch* untuk menangkap eksepsi apabila terdapat error saat pengiriman pesan. Pengiriman pesan pada Gambar 4 baris 12 dengan penjelasan bahwa pesan yang sudah difilter mengisi parameter fungsi *Pesan(pes)* dan dipanggil sehingga akan menjalankan fungsi *Pesan(pes)* pada Gambar 7 untuk proses pengiriman pesan dan dalam satu waktu menerima balasan (*response*) dari server kemudian hasilnya ditampung pada variabel *resultpesan*.

1	class ThreadPesan extends Thread {
2	private CommandListener listener;
3	
4	public void setListener(CommandListener cl) {
5	listener = cl;
6	}
7	
8	public void run() {
9	pesanTxtBox = textBoxPesan.getString();
10	pes = FilterPesan.filter(pesanTxtBox);
11	try {
12	resultpesan = Pesan(pes);
13	formReplay Pesan.deleteAll();
14	ParsingXML(resultpesan);
15	pesanText = getParXml();
16	formReplay Pesan.append(pesanText);
17	formReplay Pesan.addCommand(CmdBack_RepPes);
18	formReplay Pesan.addCommand(CmdSim_RepPes);
19	} catch (Exception e) {
20	formReplay Pesan.deleteAll();
21	formReplay Pesan.append("Jaringan sibuk / format pesan salah");
22	formReplay Pesan.addCommand(CmdBack_RepPes);
23	}
24	}
25	}

Gambar 4. Thread kirim dan terima pesan

Setelah balasan (*response*) diterima maka sebelum ditampilkan akan diparsing dengan memanggil fungsi *ParsingXML* pada Gambar 4 baris 14, untuk fungsi *ParsingXML* pada Gambar 6. Hasil parsing akan ditampung pada variabel *pesanText* Gambar 4 baris 15 dan kemudian ditampilkan, Gambar 4 baris 16.

1	public final class FilterPesan {
2	
3	public static String filter(String message) {
4	
5	if (message == null) {
6	return (null);
7	}
8	char content[] = new char[message.length()];
9	message.getChars(0, message.length(), content, 0);
10	StringBuffer result = new StringBuffer(content.length + 50);
11	for (int i = 0; i < content.length; i++) {
12	switch (content[i]) {
13	case ' ':
14	result.append("%20");
15	break;
16	case '#':
17	result.append("%23");
18	break;
19	case '&':
20	result.append("%20");
21	break;
22	case '\"':
23	result.append(""");
24	break;
25	default:
26	result.append(content[i]);
27	}
28	}
29	return (result.toString());
30	}
31	}

Gambar 5. Filter Pesan

Filter pesan pada Gambar 5 digunakan untuk menyaring pesan agar bisa dikirimkan melalui HTTP, sebab tidak semua karakter terdeteksi bisa dibaca oleh http, sebagai contoh karakter spasi dalam format http diganti %20. Pada baris 13–24 merupakan method untuk memfilter karakter agar bisa dibaca oleh http.

Gambar 6 baris 19-26 merupakan blok *listing* untuk *parsing* XML, maksud dari *listing* tersebut yaitu program hanya akan menampilkan data XML yang diawali dan diakhiri tag body, kemudian isi dari tag body akan diambil datanya dan ditampung pada variabel *items*, pada Gambar 6 baris 30. Untuk pemanggilan fungsi *parseXMLItems* yaitu pada Gambar 6 baris 6, yang masuk pada fungsi *ParsingXML*.

3.1.2 Membuka Koneksi HTTP & Mengirim Pesan

Pengiriman pesan pada aplikasi sms gateway over gprs ini dengan membuka koneksi http yang akan menghubungkan ke server router, dengan maksud bahwa alamat server dan variabel pesan yang dikirimkan akan ditampung pada String yang diisikan pada parameter fungsi untuk membuka koneksi http. Untuk lebih jelasnya bisa dilihat pada Gambar 7. Gambar 7 baris 8 merupakan *listing* untuk membuka koneksi dengan *server* yang parameter fungsi koneksinya berisi *string url* pada baris 6, *string url* ini merupakan alamat http dari *server* untuk pengiriman pesan.

1	public String ParsingXML(String xml) {
2	try {
3	ByteArrayInputStream bin = new ByteArrayInputStream(xml.getBytes());
4	InputStreamReader in = new InputStreamReader(bin);
5	XmlParser parser = new XmlParser(in);
6	parsekXMLItems(parser, xml);
7	} catch (IOException e) {
8	e.printStackTrace();
9	}
10	return xml;
11	}
12	
13	public void parsekXMLItems(XmlParser parser, String items) throws IOException {
14	variabel inItem = false;
15	while (true) {
16	ParseEvent event = parser.read();
17	switch (event.getType()) {
18	case Xml.START_TAG:
19	if (event.getName().equals("body")) {
20	inItem = true;
21	}
22	break;
23	case Xml.END_TAG:
24	if (event.getName().equals("body")) {
25	inItem = false;
26	}
27	break;
28	case Xml.TEXT:
29	if (inItem) {
30	items = event.getText();
31	setParXml(items);
32	}
33	break;
34	case Xml.END_DOCUMENT:
35	return;
36	}
37	}
38	}

Gambar 6. Parsing Pesan

1	public String Pesan(String url) throws IOException {
2	HttpConnection konpes = null;
3	int esponse;
4	
5	try {
6	url = "http://sigawe.net:8080/server/sendmessage?&pesan=" + pes;
7	
8	konpes = (HttpConnection) Connector.open(url);
9	InputStream is = konpes.openInputStream();
10	response = konpes.getResponseCode();
11	if (esponse != HttpURLConnection.HTTP_OK) {
12	throw new IOException("HTTP Error with code: " + esponse);
13	}
14	
15	int size = 100;
16	StringBuffer tmp = new StringBuffer(size);
17	InputStreamReader Input = new InputStreamReader(is);
18	char[] buffer = new char[size];
19	

20	<code>while ((Input.read(buffer, 0, buffer.length)) != -1) {</code>
21	<code> tmp.append(buffer);</code>
22	<code> buffer = new char[size];</code>
23	<code>}</code>
24	
25	<code>Input.close();</code>
26	<code>String resultText = tmp.toString();</code>
27	<code>return resultText;</code>
28	
29	<code>} catch (Exception e) {</code>
30	<code> throw new IllegalArgumentException(e.getMessage());</code>
31	<code>} finally {</code>
32	<code> if (konpes != null) {</code>
33	<code> konpes.close();</code>
34	<code> }</code>
35	<code>}</code>
36	
37	<code>}</code>

Gambar 7. Koneksi HTTP

3. HASIL DAN PEMBAHASAN

Aplikasi mobile yang dibangun telah dapat melakukan parsing dan filtering pesan. Hal ini telah diuji dengan contoh input pesan-pesan pada layanan untuk menampilkan informasi KRS dalam aplikasi yang berbasis SMS over GPRS. Contoh input dan hasil parsing ditunjukkan pada Tabel 2 berikut ini:

Tabel 2. Hasil Percobaan Filtering dan Parsing

No	Teks	Proses	Hasil
1.	REG#04018234#test#test	Filtering	REG%2304018234%23test%23test
2.	<xml> 06016007</xml>	Filtering	<xml>%2006016007</xml>
3.	&test1&'test2'&"test3"	Filtering	&test1&apos;test2' &quot;test3"
4	<xml> <body>test</body> </xml>	Parsing	test
5	<xml> <body>test</body> <body>test2</body> </xml>	Parsing	test test2

4. SIMPULAN

Setelah melakukan penelitian Aplikasi SMS *gateway* yang menggunakan konektivitas GPRS sebagai pengiriman Pesan untuk mengakses sistem informasi, maka dapat disimpulkan bahwa:

- Aplikasi mobile yang dibangun telah dapat menangani pengiriman pesan dari perangkat mobile ke webserver menggunakan metode filter untuk pesan yang dikirim dan metode parsing untuk response yang diterima dari webserver.
- SMS Over GPRS dapat digunakan sebagai media alternatif untuk mengakses data dari suatu sistem informasi melalui protokol komunikasi http dengan mekanisme request-response.

Selain gateway SMS, telah dikembangkan gateway MMS. Ini dimungkinkan karena berkembangnya jalur komunikasi data dari GPRS ke 3G. Hal ini karena berkembangnya kebutuhan dari teks ke isi multimedia seperti teks, *image*, *audio clip*, bahkan *video clip* [6].

DAFTAR PUSTAKA

- [1]. Ching-Shine, H., W. Tzoo-Shaang, et al., "**An integration Platform for Developing Digital Life Applications**", IEEE International Conference on Parallel and Distributed Systems, 2007.
- [2]. Faris Jaya Mustika, "**Membangun Server *chatting* sebagai sarana komunikasi data antar *mobile phone* menggunakan jalur GPRS**", Skripsi S1, Universitas Ahmad Dahlan, Yogyakarta, 2007.
- [3]. Ananta Wijaya, "**Aplikasi Mobile SMS Gateway Over GPRS (Studi kasus swalayan Madurejo Piyungan Yogyakarta)**", Skripsi S1, Universitas Ahmad Dahlan, Yogyakarta, 2008.
- [4]. M. Shalahuddin, Rosa A.S, "**Pemrograman J2ME**", Penerbit Informatika, Bandung.
- [5]. Siregar, Ivan Michael, "**Membangun aplikasi Chat lewat GPRS dengan J2ME menggunakan NetBeans IDE 5.0**", Gava Media, Yogyakarta, 2007.
- [6]. Lu, G., H. Ladas, et al., "**Heading for Multimedia Message Service in 3G**", 6th IEEE International Conference on 3G and Beyond, 2005.