

SINKRONISASI DATABASE TERHADAP PENGIRIMAN DATA SMS MENGGUNAKAN MULTIPLE MODEM SMSGATEWAY

Luqman Affandi ¹⁾, Dhebys Suryani ²⁾, Putra Prima A ³⁾

¹⁾ Politeknik Negeri Malang

Email: laffandi@polinema.ac.id

²⁾ Politeknik Negeri Malang

Email: dhebys.suryani@gmail.com

³⁾ Politeknik Negeri Malang

Email: putraprima@polinema.ac.id

Abstract

SMS Gateway Server will connect with a database which will be sent via sms sender modem. When a message is sent to the recipient, the time required to send the SMS is for a long time. To be able to solve these problems can be used multiple modem, so the sms delivery expense can be divided into several modems in one delivery. This requires the technique of how to manage the sms recipients in order to be well distributed to multiple modems at once. The technique that can be used is database synchronization with multiple modems, by dividing the number of messages to be sent. Such data synchronization can save about 50% of the time of delivery.

Keywords: SMS Gateway, synchronization, time of delivery

1. Pendahuluan

1.1. Latar Belakang

Pada umumnya sebuah server SMSGateway akan terhubung dengan sebuah database yang selanjutnya data akan dikirim melalui modem pengirim sms. Ketika pesan yang dikirim ke penerima dengan jumlah sedikit dalam sekali kirim, maka tidak akan terjadi kendala yang berarti. Tetapi jika jumlah penerima banyak, hal ini akan menemui kendala yang sangat berarti. Kendala yang ditemui yaitu memerlukan waktu untuk dapat mengirimkan data sms tersebut karena rata-rata pengiriman satu sms memerlukan 1 sampai 2 detik.

Untuk dapat menyelesaikan permasalahan tersebut dapat ditangani dengan pembangunan SMSGateway dengan multiple modem, sehingga beban pengiriman sms dapat dibagi ke beberapa modem dalam satu pengiriman. Hal ini memerlukan sebuah teknik bagaimana mengelola penerima sms agar dapat terdistribusi dengan baik ke beberapa modem sekaligus.

Teknik yang dapat digunakan salah satunya adalah sinkronisasi data yang ada di database. Secara *default* gammu sebagai aplikasi sms gateway mempunyai satu koneksi database. Dengan mendeteksi jumlah pesan yang akan dikirim, maka akan dapat dihitung beban pengiriman pesan yang

dilakukan oleh tiap-tiap modem. Sehingga dengan sinkronisasi tersebut dapat meningkatkan kinerja sebuah server sms gateway.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, maka permasalahan dalam penelitian ini adalah bagaimana mengimplementasikan mensinkronisasi database untuk membangun multiple modem SMSGateway untuk pengiriman sms dalam jumlah besar?

1.3. Batasan Masalah

Batasan masalah dari penelitian ini adalah:

1. Aplikasi Menggunakan dua jenis modem sms.
2. Aplikasi dibangun menggunakan aplikasi open source yaitu CodeIgniter dan Gammu yang diimplementasikan di sistem operasi Ubuntu 14.10.
3. Database yang digunakan adalah MySQL.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah terciptanya aplikasi SMSGateway yang handal agar mampu pengiriman sms dalam jumlah banyak dengan waktu yang singkat.

1.5 Manfaat penelitian

Manfaat yang ingin dicapai dari penelitian ini adalah:

1. Membantu agar informasi lebih cepat diterima.
2. Menghindari tidak terkirimnya sms.
3. Membantu admin dalam pengelolaan smsgateway.

2. Metode Penelitian

2.1 SMS Gateway

SMS Gateway adalah sebuah gerbang yang menghubungkan antara komputer dengan client melalui SMS. Client secara tidak langsung berinteraksi dengan aplikasi / sistem melalui SMS Gateway. Saat melakukan SMS, maka informasi terpenting yang diperlukan adalah nomor tujuan dan pesan, maka itulah yang sebenarnya diolah oleh *SMS Gateway*[1].

2.2 CodeIgniter

CodeIgniter adalah sebuah web application framework yang bersifat open source digunakan untuk membangun aplikasi php dinamis [2]. Tujuan utama pengembangan *CodeIgniter* adalah untuk membantu pengembang perangkat lunak untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. *CodeIgniter* menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. *CodeIgniter* diperkenalkan kepada public pada tanggal 28 februari 2006.

CodeIgniter sendiri dibangun menggunakan konsep *Model-View-Controller development pattern*. *CodeIgniter* sendiri merupakan salah satu *framework* tercepat dibandingkan dengan framework lainnya. Pada acara frOSCon (August 2008), pembuat php Rasmus Lerdorf mengatakan dia menyukai *codeigniter* karena dia lebih ringan dan cepat dibandingkan *framework* lainnya.

2.3 MySql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi

komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

2.4 Gammu

Gammu adalah sebuah aplikasi cross-platform yang digunakan untuk menjembatani/mengomunikasikan antara database SMS Gateway dengan *sms devices* [4]. Aplikasi Gammu berupa daemon yang berjalan secara background. Setiap saat, gammu memonitor sms devices dan database sms gateway. Saat ada sms masuk ke sms devices, maka gammu langsung memindahkannya ke dalam inbox dalam database sms gateway. Sebaliknya saat Aplikasi Pengirim SMS memasukkan sms ke dalam *outbox* dalam database sms gateway, maka gammu mengirimkannya melalui *sms devices*, dan memindahkan sms ke sentitem dalam database.

3. Pembahasan

3.1 Perencanaan Penelitian

Penelitian dilakukan di SMAN 1 Genteng Banyuwangi. Penelitian dilaksanakan selama 8 bulan dimulai pada bulan Maret 2017 sampai dengan Oktober 2017. Di sekolah tersebut menggunakan sms gateway untuk mengirimkan informasi ke guru, karyawan dan siswa, melalui aplikasi web dan pengiriman sms dengan format tertentu oleh admin.

3.3 Teknik pengumpulan data/instrumen penelitian

Data akademik diperlukan dalam penelitian ini, terutama data induk siswa, guru, jadwal dan absensi siswa. Data tersebut diperoleh dengan observasi dan wawancara di SMAN 1 Genteng khususnya di bidang kesiswaan.

3.2 Analisis Masalah

Gammu digunakan sebagai aplikasi perantara antara modem dengan aplikasi sms gateway. Gammu akan mengubah data sms ke dalam data yang disimpan ke dalam database, yang selanjutnya akan diolah oleh aplikasi sms gateway. Aplikasi tersebut akan mengakses database, kemudian diolah sesuai dengan kebutuhan untuk selanjutnya data dikirimkan kembali melalui modem. Semua sms masuk akan dicatat dalam tabel inbox, sedangkan sms yang akan dikirim

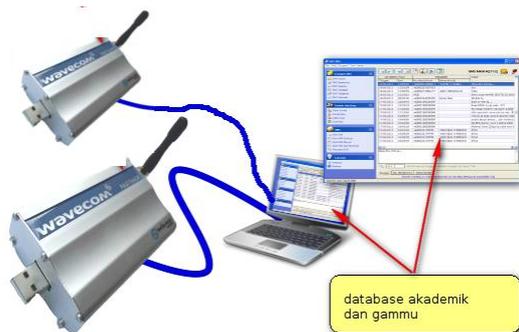
dimasukkan ke dalam tabel outbox. SMS yang telah terkirim akan dicatat dalam tabel senditem.

Permasalahan akan muncul jika terdapat penumpukan pengiriman sms di sebuah modem, hal ini akan membuat proses semakin lama. Jika server sms gateway memiliki dua atau lebih modem, seharusnya beban salah satu modem dapat didistribusikan ke semua modem agar lebih cepat proses pengirimannya. Tetapi tiap modem secara default mempunyai satu database sehingga jika menggunakan banyak modem akan memerlukan lebih dari satu database. Data *inbox* maupun *outbox* terpisah-pisah sesuai dengan database yang digunakan.

Dari masalah tersebut diperlukan sinkronisasi data *inbox* maupun *outbox* agar dapat diolah dan dibagi ke banyak modem agar dapat mempercepat proses pengiriman sms. Langkah yang dapat dilakukan yaitu:

1. Menghubungkan antar database agar terjadi sinkronisasi data khususnya tabel *inbox* dan *outbox*. Hal ini dapat dilakukan dengan perintah *join* antar database.
2. Membuat konfigurasi gammu sebanyak modem dengan id yang berbeda-beda, sehingga data dapat disimpan hanya dalam satu database. Sinkronisasi dapat dilakukan dengan memanfaatkan data id modem.

Secara garis besar sistem yang dibangun seperti pada gambar berikut:



Gambar 1. Skema Umum Database dan Multiple Modem

3.3 Desain Sistem

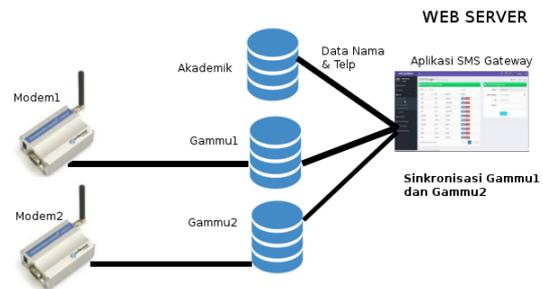
Untuk menggambarkan cara kerja sistem sinkronisasi database pada sms gateway dengan banyak modem dapat digunakan topologi sistem, *flowchart*, *Use Case* dan *Activity Diagram*.

3.3.1 Topologi Sistem

Setelah dilakukan analisis masalah, maka dapat ditentukan dua cara untuk melakukan sinkronisasi database, yaitu menggunakan banyak database dan menggunakan banyak konfigurasi gammu.

3.3.1.1 Topologi dengan Banyak Database

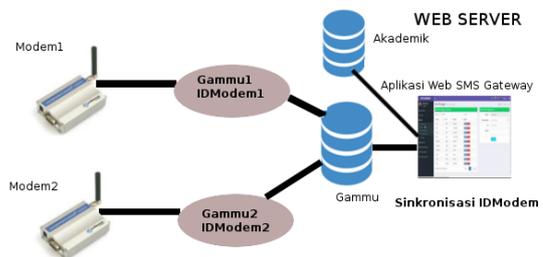
Topologi banyak modem dapat dilihat seperti pada gambar 2 berikut:



Gambar 2. Topologi Sistem Sinkronisasi dengan banyak Database

3.3.1.2 Topologi dengan Banyak Konfigurasi Gammu

Topologi banyak modem dengan beberapa konfigurasi Gammu dapat terlihat seperti pada gambar 3 berikut:



Gambar 3. Topologi Sinkronisasi Dengan Banyak Konfigurasi

3.3.2 Flowchart Sistem

Pada dasarnya proses pembagian beban pengiriman sms ke beberapa modem mempunyai prinsip yang sama yaitu menggunakan rumus rata-rata. Adapun rumus rata-rata yang dipakai sebagai berikut:

$$BM = \left\lfloor \frac{\Sigma S}{\Sigma B} \right\rfloor \dots \dots \dots 1$$

dimana:

BM = Beban sms tiap modem

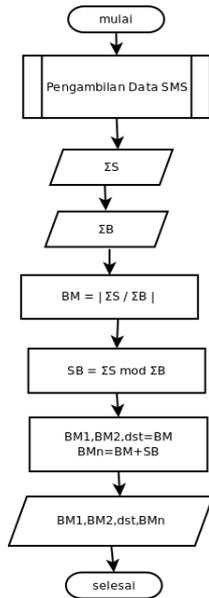
ΣS = Jumlah sms yang akan dikirim

ΣB = Jumlah modem yang dipakai.

Untuk mengatasi jumlah beban sms yang tidak sama tiap-tiap modem, maka dapat digunakan rumus sisa hasil bagi, yang selanjutnya hasil sisa bagi tersebut dibebankan ke salah satu mode. Berikut rumus sisa hasil bagi:

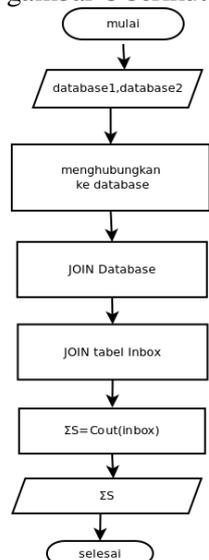
$$SB = \Sigma S \text{ mod } \Sigma B \dots\dots\dots 2$$

Dari rumus-rumus tersebut dapat dibuatkan flowchart perhitungan beban sms tiap modem seperti terlihat pda gambar 4 berikut:

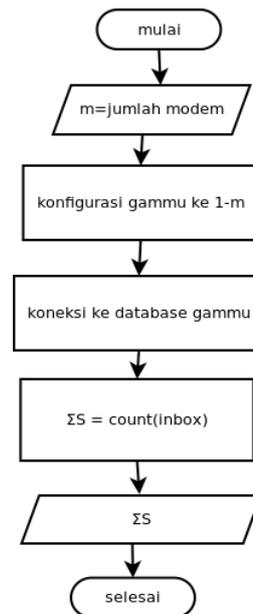


Gambar 4. Flowchart Perhitungan Beban

Perbedaan algoritma terjadi pada saat sinkronisasi data untuk perhitungan jumlah sms yang akan dikirim seperti terlihat pada gambar 5 dan gambar 6 berikut:



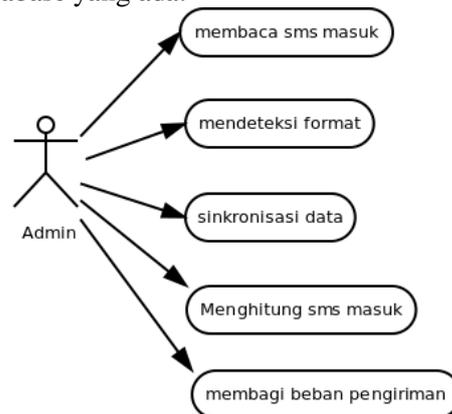
Gambar 5. Sinkronisasi Antar Database



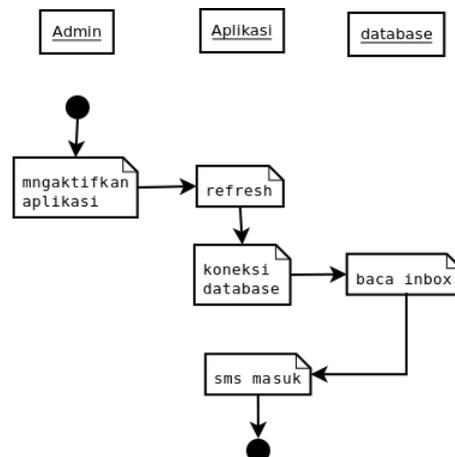
Gambar 6. Sinkronisasi Data Inbox

3.3.3 Use Case

Aplikasi web sms gateway merupakan sarana untuk mensinkronisasikan database yang ada.



Gambar 7 Use Case Aplikasi



Gambar 8. Activity Diagram Pembacaan SMS Masuk

3.3.4 Activity Diagram

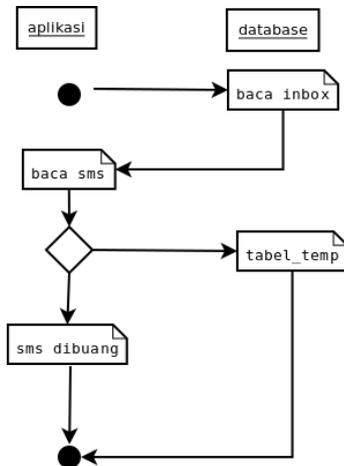
Diagram ini digunakan untuk menjelaskan bagaimana langkah dari tiap proses yang ada di *use case*.

3.3.4.1 Activity Diagram Pembacaan SMS Masuk

Gambar 8 merupakan activity diagram untuk melihat sms yang masuk di dalam database:

3.3.4.2 Activity Diagram Pengecekan Format

Activity ini menunjukkan proses pendeteksian format sms yang dikirim oleh admin untuk mengirimkan pesan ke penerima.



Gambar 9. Activity Diagram Pengecekan Format SMS

4.1 Implementasi Aplikasi

Setelah dilakukan analisis dan desain, selanjutnya adalah proses implementasi program. Implementasi terbagi menjadi dua bagian yaitu penerapan database dan implementasi aplikasi web sms gateway.

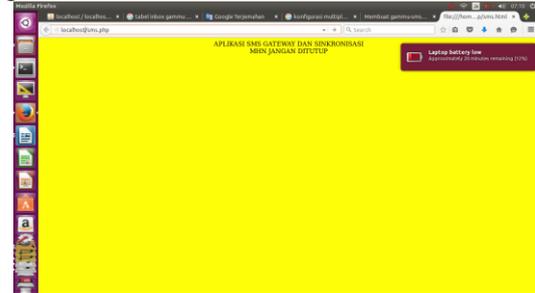
4.1.1 Implementasi Database

Dalam implementasi ini terdapat dua model database, yaitu menggunakan satu database dan 2 database. Untuk implementasi dengan satu database mengacu pada format dan struktur database gammu agar dapat dihubungkan dengan aplikasi gammu. Struktur tabel gammu sebagai berikut [7]:

- Gammu: tabel konfigurasi Gammu
- inbox: tabel tempat penyimpanan sms masuk.
- Outbox: tabel tempat sms keluar
- outbox_multipart: tabel untuk sms yang mempunyai banyak halaman

- phone: tabel untuk menyimpan nomer telephon
 - senditems: tabel untuk menyimpan pesan yang telah atau gagal dikirim
- Setelah dilakukan instalasi database, maka langkah kedua adalah melakukan instalasi gammu.

Setelah dilakukan konfigurasi tersebut, langkah berikutnya adalah menjalankan aplikasi sms gateway dan melakukan sinkronisasi seperti terlihat pada gambar 10 berikut:



Gambar 10. Aplikasi sms Gateway dan Sinkronisasi Database

Setelah terdapat sms masuk, akan terlihat perbedaan antara metode 2 database dan 1 database. Hasil dari satu database seperti berikut:

ID	SenderNumber	Coding	UDH	SMSNumber	Class	TextDecoded	ID	RecipientID	Processed
00	+62395962221	Default_No_Compression		+6239500000	-1	info apnd1	40	modem1	false
	+6239524144909	Default_No_Compression		+6281100000	-1	Block BR501	46	modem1	true
	+62395962221	Default_No_Compression		+6239500000	-1	info 38207	48	modem1	true
	+6239524144909	Default_No_Compression		+6281100000	-1	Block BR503	39	modem1	true
	+6239524144909	Default_No_Compression		+6281100000	-1	Block BR503	37	modem1	true
	+6239524144909	Default_No_Compression		+6281100000	-1	Block BR503	36	modem1	true
	+6239524144909	Default_No_Compression		+6281100000	-1	Block BR503	35	modem1	true
	+62395962221	Default_No_Compression		+6239500000	-1	Block Inq1	34	modem1	true
	+6239524144909	Default_No_Compression		+6281100000	-1	INFO BR001	32	modem1	true
	+6239524144909	Default_No_Compression		+6281100000	-1	INFO BR003	33	modem1	true
	+62395962221	Default_No_Compression		+6239500000	-1	info Inq2	31	modem1	true
	+62395962221	Default_No_Compression		+6239500000	-1	info Inq1	30	modem1	true
	+62395962221	Default_No_Compression		+6239500000	-1	info Inq1	29	modem1	true
	+6239524144909	Default_No_Compression		+6281100000	-1	INFO BR003	28	modem1	true

Gambar 11 Hasil Konfigurasi 2 database

Pada gambar 11 terlihat bahwa semua IDModem tertulis Modem1 sesuai dengan penamaan pada konfigurasi Gammu. Gambar 12 menunjukkan hasil database yang menggunakan konfigurasi satu database dengan banyak konfigurasi gammu:

TextDecoded	ID	recipientID
Maaf pak tadi saya telpon tdk dia	894	Phone1
arena tdk ada trn yg memberitah	895	Phone1
Ok.	896	Phone1
Jadikan hits baru dr ELLO dik, Bul	897	Phone1
Gmn pak, uda dikrim?	898	Phone1
Ok, username a4nd0ank,makash	899	Phone1
Ok,thx	900	Phone1
Ok,pak,nanti saya cb,makash	901	Phone1
Tgl 26 Jan 2010, bca, rp 670.123,	902	Phone1
Mbak td telp ada apa?aq lg turu a	903	Phone2
lIn raya BEKASI tp0214711110,0	904	Phone2
>Transaksi Ditolak.Perintah trans:	905	Phone2
Informasi saldo xxx011. Tgl 25-01	906	Phone2
Saldo rek. Anda pada tgl 25-01-11	907	Phone2
Test	908	Phone1

Gambar 12 Tabel Inbox dengan Konfigurasi Banyak Modem

Pada Gambar 12 Terlihat bahwa IdModem terdiri atas Phone1 dan Phone 2 sesuai dengan modem penerima sms dari pengguna.

Setelah mendapatkan data inbox, langkah selanjutnya dilakukan proses Join Database dengan perintah sebagai berikut:

```
SELECT *
```

```
FROM A.inbox t1 JOIN B.inbox t2
```

```
ON t2.idphone = t1.idphone;
```

dimana A adalah database Gammu1 dan B adalah Gammu2. Sedangkan untuk mendapatkan jumlah sms masuk dapat digunakan perintah count seperti berikut:

```
Select count(*) as jml=sms from gabungan
```

4.2 Pengujian Sistem

Setelah proses implementasi dilakukan, maka langkah selanjutnya adalah pengujian terhadap kecepatan pengiriman sms dari hasil sinkronisasi database. Tabel 1 menunjukkan waktu pengiriman dengan beberapa banyak beban sms.

Tabel 1. Pengujian Tanpa Menggunakan Sinkronisasi

No.	Jumlah Beban SMS	Waktu (detik)
1.	10	15
2.	20	38
3.	25	50
4.	50	105
5.	60	130
	Rata-rata pengiriman	

Dari tabel 1 terlihat bahwa rata-rata 2,05 detik tiap pengiriman sms. Tabel 2 menunjukkan hasil pengujian menggunakan sinkronisasi database.

Tabel 2. Pengujian Menggunakan Sinkronisasi

No.	Jumlah Beban SMS	Modem 1 (Detik)	Modem 2 (Detik)	Rata-Rata Pengiriman (Detik)
1.	10	10	9	9,5
2.	20	15	19	17
3.	25	35	35	35
4.	50	50	54	52
5.	60	34	35	34,5

Dari hasil pengujian di atas terlihat bahwa hasil pengiriman dengan memanfaatkan sinkronisasi database dapat membagi beban pengiriman modem sampai kurang lebih 50%. Demikian pengiriman pesan ke penerima akan semakin cepat.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil analisis, pengujian dan hasil, dapat disimpulkan bahwa :

1. Database gammu dapat disinkronisasikan dengan perintah Join antar database.
2. Penggunaan banyak modem memungkinkan dilakukan dengan dua cara yaitu dengan banyak database dan banyak konfigurasi gammu.
3. Sinkronisasi data dapat menghemat kurang lebih 50% waktu pengiriman.

5.2 Saran

Untuk menjaga kestabilan sistem, maka saran yang dapat dilakukan adalah diperlukan pendeteksian berapa jumlah pulsa di tiap-tiap modem agar proses pengiriman dapat berjalan dengan lancar.

6. DAFTAR PUSTAKA

- [1] Ramadhika, "SMS Gateway Menggunakan Gammu dan MySQL". http://www.ubaya.ac.id/2014/content/articles_detail/33/SMS-Gateway-menggunakan-Gammu-dan-MySQL.html
- [2] CodeIgniter, "About CodeIgniter", www.codeigniter.com
- [3] Oracle, "MySQL". www.mysql.com/
- [4] Cihar, Michal. "Installing Gammu". <http://wammu.eu/gammu/>
- [5] David M. Kroenke, 2005:60, "Unified Modeling Language"
- [6] Rosa A.S, M. Shalahuddin, 2013:140, "Unified Modeling Language"
- [7] Wammu, "Gammu Tables", wammu.eu/docs/manual/smsd/tables.html