

Sinusoida

Jurnal Penelitian Dan Pengkajian Elektro

DAFTAR ISI

		Halaman
1.	Rancang Bangun Sistem Transportasi Booking Berbasis Sms Gateway Menggunakan Software GAMMU Edy Supriyadi dan Ayu Zuafah	1 – 10
2.	Rancangan Antena Panel Mikrostrip 8 Larik Pada Aplikasi Wimax Frekuensi 2,4 Ghz Aditya Indra P dan Heru Abrianto	11 – 19
3.	Evaluasi Kualitas Jaringan Serat Optik DWDM 40λ Melalui Perhitungan Dan Pengukuran Untuk Perbaikan Kinerja Arfian ahmad dan Zaqqy Kevin Laison	20 – 27
4.	Perancangan Dan Implementasi Multicast Routing Pada Trafik Video Streaming Untuk Mengatasi Keterbatasan Bandwidth Mufti Gafar dan Yuli Yektiani	28 – 36
5.	Perancangan Sistem Kendali Lampu Dan AC Pada <i>Network Operation Center</i> ISTN Melalui Wifi Web Server Berbasis Arduino Uno Rachman Soleman, M. Febriansyah dan Mohammad Rizardi	37 – 47
6.	Metode <i>Low Latency Queuing</i> Untuk Mengurangi Terjadinya Kongesti Data Pada Jaringan IP Febriansyah dan Irmayani	48–57
7.	Implementasi GZIP Compression Untuk Mempercepat Pengiriman Data Di Jaringan WAN Eyasyer Hendra dan Mohammad Hamdani	58 – 64
8.	Perencanaan Instalasi Listrik Untuk Seksi Pengecatan di Industri Otomotif Sugianto, A. Muis dan Arief Priambud	65 – 77
9.	Pengaturan Kecepatan Motor Induksi Fasa Tiga Untuk Menjaga Tekanan Tetap Pada Pipa Outlet Motor Pompa Injektor Chlorine Mohammad Amir dan Adis Suhur Nahar	78 – 91
10.	Perbaikan Performansi <i>Video Call</i> Menggunakan Jaringan LTE H. Budihardjo Gozali dan Arnold Julyus.S	92 – 105
11.	Sistem Keamanan Parkir Berbasis RFID dan Kamera Surya Alimsyah dan Eko Subiyantoro	106 - 118

RANCANG BANGUN SISTEM TRANSPORTASI BOOKING BERBASIS SMS GATEWAY MENGGUNAKAN SOFTWARE GAMMU

Edy Supriyadi¹, Ayu Zuafah²

^{1,2}Program Studi Teknik Elektro, Fakultas Teknik Industri,
Institut Sains Dan Teknologi Nasional Jakarta



ABSTRAK

PT. Indodev Niaga Internet (DataOn) sudah menerapkan aplikasi Internal System untuk transportasi booking ke klien dengan pemanfaatan teknologi informasi dan teknologi komputer yang dapat memudahkan segala aspek bidang pekerjaannya. Saat ini untuk melakukan transportasi booking, para karyawan harus mengakses sistem terlebih dahulu melalui PC. Pada penelitian ini dibuat sebuah aplikasi untuk mempermudah karyawan dalam mengakses sistem tersebut, yaitu pembuatan Sistem Transportasi Booking Berbasis SMS Gateway Menggunakan Software Gammu. Karyawan melakukan transportasi booking dengan cara mengirimkan SMS dengan format yang telah ditentukan. Konfirmasi jenis kendaraan akan dikirimkan oleh sistem secara otomatis kepada karyawan melalui SMS. Selain itu, notifikasi juga dikirimkan apabila format SMS salah. Untuk koneksi SMS gateway menggunakan modem Huawei E153. Pengujian dilakukan dengan pengcapturean data menggunakan software Wireshark. Parameter QoS yang dianalisis berupa throughput, delay, dan packet loss. Hasil analisa data dari percobaan yang dilakukan menunjukkan bahwa pada saat SMS diterima oleh sistem diperoleh delay rata-rata sebesar 231,5 ms, packet loss yang bernilai 0%, sedangkan nilai throughputnya akan semakin turun seiring dengan bertambahnya jumlah paket.

Kata Kunci : SMS Gateway, Gammu, Wireshark, Throughput, Delay, Packet Loss

ABSTRACT

PT. Indodev Internet Commerce (DataOn) has adopted the Internal System application for booking transport to the client with the use of information technology and computer technology that can facilitate all aspects of the field work. Currently to perform transport booking, the employee must first access the system through a PC. In this study made an application to facilitate employees to access the system, namely the manufacture of Transport Booking System Based SMS Gateway Using Software Gammu.

Employees do transport booking by sending an SMS with a predefined format. Confirm the type of vehicle to be delivered by the system automatically to employees via SMS. In addition, a notification is also sent if the SMS format is incorrect. SMS gateway connection using a modem Huawei E153. Tests conducted by capturing data using Wireshark software. QoS parameters are analyzed such as throughput, delay, and packet loss. Results of analysis of data from experiments conducted show that when an SMS is received by the system obtained average delay of 231.5 ms, packet loss is 0%, while the throughput rate will fall as the number of packets.

Keywords: SMS Gateway, Gammu, Wireshark, Throughput, Delay, Packet Loss

1. PENDAHULUAN

Perkembangan teknologi telekomunikasi dan informasi meningkat dengan sangat maju seiring dengan perkembangan teknologi komputer yang terus meningkat dalam hitungan hari. Saat ini hampir semua jenis lembaga, yaitu baik lembaga profit (perusahaan) maupun non profit (universitas, sekolah, training) telah memanfaatkan pengembangan teknologi informasi untuk meningkatkan manajemen dan kinerjanya melalui suatu sistem informasi online, sistem penunjang keputusan, sistem pakar, dan lain sebagainya. Lembaga profit, contohnya PT. Indodev. Niaga Internet (DataOn), saat ini sudah menerapkan aplikasi Internal System untuk pemesanan

kendaraan (transportasi booking) ke klien dengan pemanfaatan teknologi informasi dan teknologi komputer yang dapat memudahkan segala aspek bidang pekerjaannya.

Saat ini, untuk melakukan pemesanan kendaraan, para karyawan harus mengakses sistem terlebih dahulu melalui PC, membuka web browser, lalu memasukkan alamat tertentu untuk mengaksesnya. Hal ini cukup merepotkan apabila ada klien yang mengharuskan karyawan untuk segera datang ke kantornya. Akan memakan waktu apabila karyawan harus melakukan pemesanan melalui PC. Dari permasalahan tersebut, maka timbul ide untuk membuat suatu aplikasi SMS Gateway untuk pemesanan kendaraan (transportasi

booking) ke klien sehingga memberi kemudahan bagi karyawan dalam melakukan pengaksesan.

Karyawan PT. Indodev Niaga Internet (DataOn) sudah menggunakan *smartphone* sebagai sarana telekomunikasi mereka sehingga aplikasi SMS Gateway ini tidak lagi menggunakan perintah AT Command yang hanya support untuk ponsel tertentu saja, melainkan menggunakan *software* Gammu karena *software* ini *compatible* dengan berbagai jenis *device* atau *smartphone*. Selain itu, Gammu merupakan aplikasi *open source* yang bisa dijalankan di Windows maupun Linux.

2. DASAR TEORI

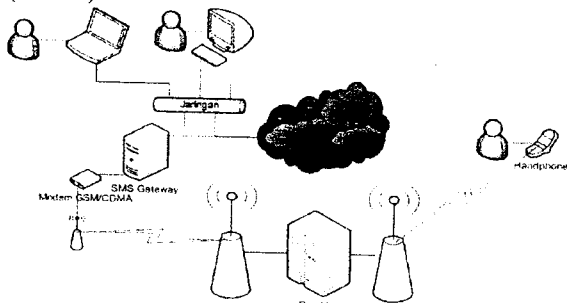
2.1 DataOn Internal Sistem

DataOn Internal Sistem adalah sistem internal yang digunakan oleh karyawan perusahaan PT. Indodev Niaga Internet (DataOn) untuk melakukan kegiatan yang menunjang pekerjaan seperti pemesanan kendaraan ke klien, pemesanan makan siang, dan dll. Saat ini, untuk melakukan pemesanan kendaraan para karyawan harus mengakses sistem terlebih dahulu melalui PC, membuka *web browser*, lalu memasukkan alamat tertentu untuk mengaksesnya.

2.2 SMS Gateway

SMS Gateway adalah sebuah program yang mengkomunikasikan sistem operasi jaringan komputer dengan perangkat komunikasi yang terpasang untuk mengirim atau menerima SMS, yaitu modem GSM/CDMA, dengan memanfaatkan teknologi seluler.

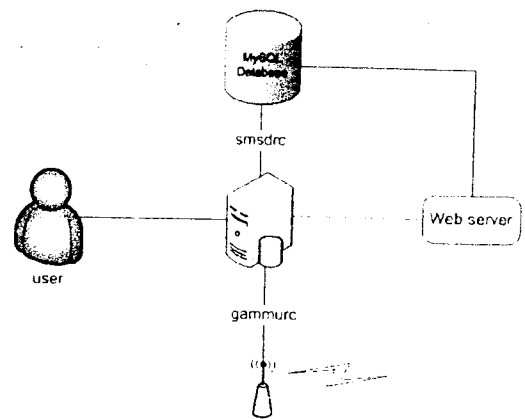
Dalam sistem SMS, mekanisme utama yang dilakukan dalam sistem adalah melakukan pengiriman pesan dari modem GSM/CDMA ke *handphone* pelanggan atau sebaliknya. Hal ini dapat dilakukan karena sebuah entitas dalam sistem seluler yang bernama *Short Message Service Center* (SMSC).



Gambar 2.1. Ilustrasi Aplikasi SMS Gateway

2.3 Gammu

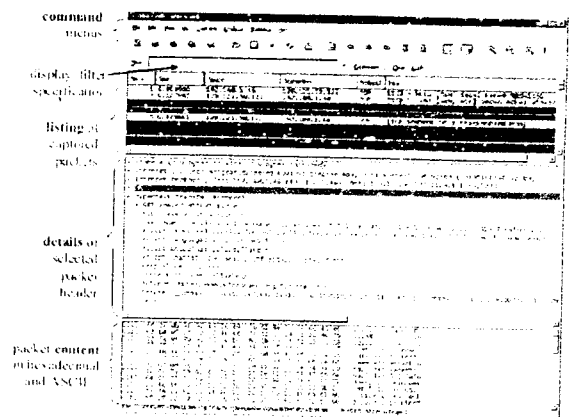
Gammu adalah suatu *software* yang ditujukan untuk membangun alikasi, *script* dan *driver* yang dapat digunakan untuk menjalankan semua fungsi yang memungkinkan pada telepon selular atau alat sejenisnya (SMS Gateway).



Gambar 2.2 Cara Kerja Gammu dalam sistem SMS Gateway

2.4 Wireshark

Wireshark merupakan salah satu dari sekian banyak *tool Network Analyzer* yang banyak digunakan oleh *network administrator* untuk menganalisa kinerja jaringannya termasuk protokol didalamnya.



Gambar 2.3 Struktur Wireshark

2.5 Throughput, Delay dan Packet Loss

• Throughput

Throughput adalah kecepatan rata-rata data yang diterima oleh suatu *node* dalam selang waktu pengamatan tertentu. Nilai *throughput* sistem ditentukan dengan Persamaan 2.1.

$$\text{Throughput} = \frac{\text{Paket data yang diterima (bit)}}{\text{Time between first \& last packets (s)}} \quad (2.1)$$

Dimana :

- paket data yang diterima adalah jumlah *bit* data dikirim dikurangi jumlah *bit* data *error*.
- *Time between first and last packet* adalah waktu paket yang dikirim oleh terminal menuju terminal lainnya.

• Delay

Delay atau waktu paket di dalam sistem adalah waktu sejak paket tiba ke dalam sistem sampai paket selesai ditransmisikan. Tabel 2.1 adalah tabel parameter kualitas sebuah jaringan dilihat dari besarnya *delay* menurut ITU-T G.114.[10]

Tabel 2.1 Standarisasi ITU-T G.114 untuk *Delay*

Kategori <i>Delay</i>	Besar <i>Delay</i>
Excellent	< 150 ms
Good	150 s d 300 ms
Poor	300 s d 450 ms
Unacceptable	> 450 ms

Sedangkan untuk menghitung *delay* dapat dicari dengan menggunakan Persamaan 2.2.

$$\text{Delay} = \frac{\text{Time between first \& last packets}}{\text{jumlah paket}} \text{ (s)} \quad (2.2)$$

Dimana :

- *Delay* = *delay* yang dikirim oleh *client* menuju *client* lainnya.
- *Time between first and last packet* = waktu paket yang dikirim oleh *client* menuju *client* lainnya
- Jumlah paket = jumlah paket data yang difilter oleh *software* wireshark

• **Packet Loss**

Packet loss adalah perbandingan seluruh paket IP yang hilang dengan seluruh paket IP yang dikirimkan antara pada *source* dan *destination*. Tabel 2.2 menunjukkan parameter kualitas sebuah jaringan dilihat dari besarnya *packet loss* menurut ITU-T G.114.[10]

Tabel 2.2 Standarisasi ITU-T G.114 untuk *Packet Loss*

Kategori Degradasi	<i>Packet loss</i>
Sangat bagus	0%
Bagus	3%
Sedang	15%
Jelek	>25%

Untuk menghitung *Packet Loss* dapat menggunakan Persamaan 2.3.

$$\text{Packet loss} = \frac{\text{Paket data dikirim} - \text{paket data diterima}}{\text{paket data dikirim}} \times 100\% \quad (2.3)$$

Dimana :

- Paket data dikirim = Jumlah paket yang dikirim
- Paket data diterima = Jumlah paket yang diterima

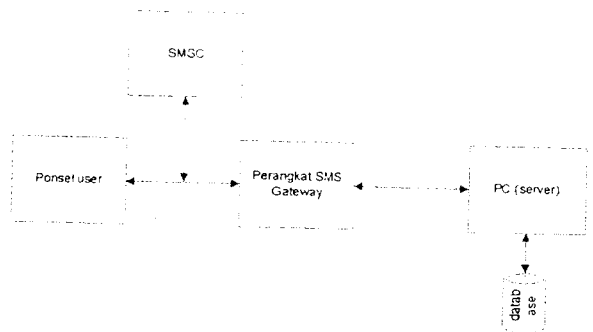
3. PERENCANAAN DAN REALISASI SISTEM

3.1 Deskripsi Sistem

Rancang bangun sistem transportasi *booking* berbasis *SMS Gateway* menggunakan *software* Gammu digunakan untuk melakukan pemesanan kendaraan karyawan untuk pergi ke

klien. Karyawan melakukan pemesanan kendaraan dengan cara mengirimkan SMS dengan format yang telah ditentukan ke *Data On Internal System* yang ada di PC server. Biasanya transportasi *booking* dilakukan satu hari sebelum tanggal keberangkatan (H-1). Jika mobil kantor tersedia maka perjalanan ke klien akan menggunakan fasilitas mobil kantor atau jika mobil kantor sudah *full booked* maka karyawan akan diberikan *voucher* taksi. Jumlah mobil kantor yang tersedia sebanyak 5 mobil dengan kapasitas maksimal 6 orang/mobil. Pembagian mobil dilakukan sesuai dengan lokasi tujuan. Konfirmasi jenis kendaraan tersebut akan dikirimkan oleh sistem kepada karyawan melalui SMS. Selain itu, karyawan juga dapat menerima notifikasi apabila format SMS yang dikirimkan salah.

Interface SMS Gateway yang digunakan adalah modem Huawei E153 yang tersambung langsung ke PC Server. Selanjutnya dengan menggunakan *software* Gammu format SMS akan dibaca kemudian diproses di dalam *database*. Untuk menghitung *delay*, *packet loss*, dan *throughput* saat SMS dikirim atau diterima maka pada server digunakan *software* Wireshark. Gambar 3.1. menunjukkan diagram blok sistem.



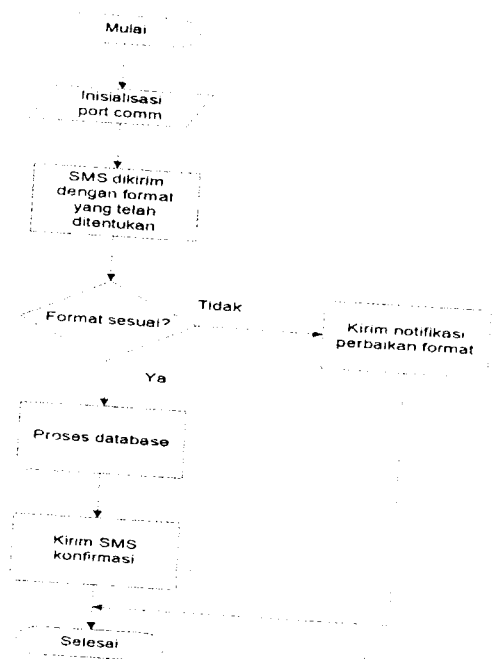
Gambar 3.1 Diagram Blok Sistem

3.2 Diagram Alir Program

Diagram alir atau *flowchart* dibuat untuk menggambarkan suatu sistem agar mudah dipahami dalam perencanaan program. Sistem transportasi *booking* berbasis *SMS Gateway* menggunakan *software* Gammu yang dibuat secara keseluruhan, seperti dapat dilihat pada Gambar 3.2.

Diagram alir program diawali dengan proses inialisasi *port com* modem yang terhubung ke komputer atau server, proses ini sangat penting karena jika setting Gammu tidak sesuai dengan *port com* modem yang telah terpasang maka sistem *SMS Gateway* tidak dapat berfungsi dengan baik. Selanjutnya karyawan mengirimkan SMS dengan format yang telah ditentukan, lalu program akan melakukan verifikasi apakah format tersebut sesuai atau tidak. Jika tidak, program akan mengirimkan notifikasi perbaikan format SMS kepada karyawan. Jika ya, program akan melakukan proses di

database selanjutnya mengirimkan SMS konfirmasi jenis kendaraan kepada karyawan.



Gambar 3.2 Flowchart Sistem SMS Gateway

3.3 Perencanaan Dan Realisasi Database SMS

Database SMS menggunakan MySQL terdiri dari 4 tabel yang dibuat untuk dihubungkan dengan sistem transportasi booking berbasis SMS Gateway yaitu:

1. Tabel *Phone*, tabel ini berisi data master dari perangkat modem yang digunakan, yaitu Huawei E153. Tabel *Phone* terdiri dari *field ID* sebagai *primary key*, *UpdatedInDB*, *InsertIntoD* dan *TimeOut* berupa informasi kapan database dibuat, *Send* dan *Receive* dibuat *Yes* agar modem bisa digunakan untuk mengirim maupun menerima SMS, *IMEI (International Mobile Equipment Identity)* berisi nomor identifikasi modem, *Client* berisi informasi versi *software* Gammu serta jenis komputer yang digunakan, *Battery* berisi informasi baterai modem, *Signal* berisi informasi kekuatan sinyal modem, serta *Sent* dan *Received* berisi total SMS yang sudah terkirim dan diterima.

Tabel 3.1 Tabel *Phone*

ID	UpdatedInDB	InsertIntoDB	TimeOut	Send	Receive	IMEI	Client	Battery	Signal	Sent	Received
1	2014-07-02 22:47:01	2014-07-02 22:38:41	2014-07-02 22:47:01	yes	yes	3544294400097	Gammu 1.25.9 Windows		5	48	7

2. Tabel *Inbox*, tabel ini berisi data SMS yang masuk. Tabel *inbox* terdiri dari *field ID* sebagai *primary key*, *UpdatedInDB* yaitu tanggal proses di database, *ReceivingDateTime* yaitu tanggal SMS diterima, *Text* yaitu pesan SMS yang diterima dalam kode enkripsi, *SenderNumber* yaitu informasi nomor telepon pengirim SMS, *SMSCNumber* adalah nomor

SMSC dari provider pengirim, *TextDecoded* yaitu pesan SMS yang diterima setelah didekripsi, *RecipientID* yaitu ID modem penerima, dan *InsertIntoDB* yaitu tanggal dimana data diproses ke Internal Sistem DataOn.

Tabel 3.2 Tabel *Inbox*

ID	UpdatedInDB	InsertIntoDB	Text	SenderNumber	SMSCNumber	TextDecoded	RecipientID	InsertIntoDB
1	2015-01-24 13:25:59	2015-01-22 11:23:58	0514063206116	428121892007	428121892007	REG TRANSPORT 20070815 SUKOHARJO	Modem1	2015-01-22 12:30:59
2	2015-01-24 13:25:59	2015-01-23 21:38:31	0514063206116	428121892007	428121892007	REG TRANSPORT 20070815 SUKOHARJO	Modem1	2015-01-23 13:25:59
3	2015-01-24 13:25:59	2015-01-23 21:38:47	0514063206116	428121892007	428121892007	REG TRANSPORT 20070815 SUKOHARJO	Modem1	2015-01-23 13:25:59
4	2015-01-24 13:25:59	2015-01-23 21:38:47	0514063206116	428121892007	428121892007	REG TRANSPORT 20070815 SUKOHARJO	Modem1	2015-01-23 13:25:59
5	2015-01-24 13:25:59	2015-01-23 21:38:47	0514063206116	428121892007	428121892007	REG TRANSPORT 20070815 SUKOHARJO	Modem1	2015-01-23 13:25:59

3. Tabel *Outbox*, tabel ini berisi data SMS yang sedang dikirim. Tabel *outbox* terdiri dari *field ID* sebagai *primary key*, *UpdatedInDB* dan *InsertIntoDB* yaitu tanggal proses update dan insert di database, *SendingDateTime* yaitu tanggal SMS dikirim, *Text* yaitu pesan SMS yang dikirim dalam kode enkripsi, *DestinationNumber* yaitu informasi nomor telepon tujuan, *TextDecoded* yaitu pesan SMS yang dikirim setelah didekripsi, *SenderID* yaitu ID modem pengirim, *SendingTimeOut* yaitu waktu *timeout* pengiriman SMS dan *CreatorID* yaitu versi *software* Gammu yang digunakan.

Tabel 3.3 Tabel *Outbox*

ID	UpdatedInDB	InsertIntoDB	Text	DestinationNumber	TextDecoded	SenderID	SendingTimeOut	CreatorID
1	2015-02-25 17:02:21	2015-02-25 17:02:21	1251206	08121892007	REG TRANSPORT 1512-2014	Modem1	2015-02-25 17:02:45	Gammu 1.25.9

4. Tabel *SentItems*, tabel ini berisi data SMS yang sudah berhasil dikirim. Tabel *SentItems* terdiri dari *field ID* sebagai *primary key*, *UpdatedInDB* dan *InsertIntoDB* yaitu tanggal proses dalam database, *SendingDateTime* yaitu tanggal dan waktu terkirimnya pesan SMS, *Text* yaitu pesan SMS yang diterima dalam kode enkripsi, *DestinationNumber* yaitu informasi nomor telepon tujuan, *SMSCNumber* adalah nomor SMSC dari provider tujuan, *TextDecoded* yaitu pesan SMS yang diterima setelah didekripsi, *SenderID* yaitu ID modem pengirim, *SequencePosition* yaitu nomor urutan proses pengiriman pesan SMS, *Status* berisikan informasi status pesan SMS yaitu pengiriman berhasil tanpa adanya laporan, dan *CreatorID* berisi informasi versi *software* Gammu yang digunakan.

Tabel 3.4 Tabel *SentItems*

ID	Sender	Receiver	Text	Time	Status
1	1234567890	0987654321	Test message	2008-01-01 10:00:00	1
2	1234567890	0987654321	Test message	2008-01-01 10:00:00	1
3	1234567890	0987654321	Test message	2008-01-01 10:00:00	1
4	1234567890	0987654321	Test message	2008-01-01 10:00:00	1
5	1234567890	0987654321	Test message	2008-01-01 10:00:00	1

3.4 Konfigurasi Software Gammu untuk Aplikasi SMS Gateway

Berikut adalah langkah-langkah yang dilakukan untuk membuat aplikasi SMS Gateway dengan software Gammu :

3.4.1 Setting GammURC

Setelah file Gammu di-install, langkah berikutnya adalah setting konfigurasi. Setting ini diperlukan guna keperluan penyesuaian jenis modem dan nomor port yang akan digunakan. Caranya adalah meng-edit file "GammURC" yang ada di dalam direktori "C:\gammu" kemudian dimasukan parameter port dan connection.

Parameter port diisi dengan nomor port sesuai lokasi colokan modem. Untuk mengetahui nomor port modem di komputer dengan masuk ke Control Panel - Device Manager - Modems. Parameter connection digunakan untuk memberi keterangan jenis koneksi yang digunakan modem untuk berkomunikasi dengan PC. Setiap jenis modem memiliki jenis koneksi yang berbeda-beda. Untuk modem Huawei E153 memiliki jenis koneksi at115200.

3.4.2 Tes Koneksi Gammu Dengan Modem

Tes koneksi Gammu dengan modem bertujuan untuk mengecek koneksi kedua perangkat tersebut agar dapat berjalan dengan baik. Caranya adalah dengan masuk folder "C:\gammu" pada direktori Gammu melalui DOS PROMPT. Lalu diketikkan perintah *Gammu Identify* pada *command prompt*. Apabila setelah perintah tersebut di-enter lalu muncul informasi mengenai modem, maka koneksi antara Gammu dengan modem sukses.

3.4.3 Setting SMSDR (Setting Konfigurasi Untuk SMS Daemon)

Langkah ini dilakukan setelah setting database untuk Gammu. SMS daemon pada Gammu digunakan untuk proses pembacaan otomatis SMS yang diterima lalu disimpan ke database. SMS daemon juga diperlukan untuk keperluan pengiriman SMS. Untuk melakukan setting SMS daemon, dibuka file "SMSDR" yang terletak di direktori 'C:\gammu'. Kemudian dimasukan parameter-parameter sebagai berikut :

- Username = nama user yang mengoperasikan database MySQL
- Password = password yang digunakan oleh database MySQL. Apabila tidak

menggunakan password, maka tidak perlu diisi.

- Database = nama database yang digunakan oleh Gammu
- Port = nomor port sesuai yang ada di file GammURC
- Connection = diisi jenis connection sesuai yang ada di GammURC
- ID Phone = ID modem yang digunakan oleh Gammu

3.4.4 Membuat dan Menjalankan Service Gammu

Langkah ini bertujuan untuk membuat service Gammu di Windows. Dengan dibuatnya service di Windows maka lebih mudah menjalankannya tanpa menggunakan perintah tertentu di *command prompt*. Untuk membuat Gammu service di Windows, dengan mengetikkan perintah *gammu-smsd -c smsdrc -i* di DOS PROMPT setelah masuk ke folder "C:\gammu". Bila ada konfirmasi bahwa proses pembuatan service sukses, maka akan terlihat service Gammu muncul pada daftar service yang berjalan di Windows. Service ini bisa dilihat melalui Control Panel Administrative Tools - Services. Untuk menjalankan service Gammu ini tinggal diklik START pada service Gammu tersebut.

3.5 Program Aplikasi SMS Gateway

Setelah membangun aplikasi SMS Gateway dengan membuat database dan konfigurasi Gammu maka selanjutnya yang perlu dilakukan adalah membuat program untuk mengintegrasikan aplikasi SMS Gateway tersebut pada DataOn Internal System untuk transportasi booking. Adapun programnya terbagi menjadi 2 bagian, yaitu program untuk menerima SMS dan mengirim SMS.

3.5.1 Menerima SMS

Apabila service Gammu masih berjalan, SMS yang diterima tidak akan masuk ke inbox ponsel karena langsung dimasukkan ke tabel inbox di database MySQL. Nomor pengirim terdapat pada field *SenderNumber* dan isi SMS terdapat pada field *TextDecoded*. Begitu juga jika sebelumnya inbox ponsel yang digunakan telah berisi SMS, maka SMS tersebut akan masuk ke tabel inbox di database MySQL, sehingga inbox ponsel kosong.

Program penerima SMS ini bertujuan untuk membaca data SMS dari database MySQL kemudian akan dimasukkan ke database DataOn Internal System. Ketika ada SMS masuk maka otomatis program ini akan membaca tabel inbox dan memasukkan datanya ke database DataOn Internal System dan selanjutnya akan diproses oleh sistem. Selain itu program ini juga akan melakukan verifikasi terhadap format SMS yang dikirim, jika

tidak sesuai dengan format yang telah ditentukan maka program memberikan perintah untuk mengirimkan SMS notifikasi kepada karyawan dengan isi pesan yaitu "Format SMS Anda salah, format SMS : REG TRANSPORT TANGGAL(DD/MM/YYYY) TUJUAN".

3.5.2 Mengirim SMS

Program ini bertujuan untuk memasukkan data dari *database* DataOn *Internal System* ke *database* MySQL untuk selanjutnya dikirimkan kepada karyawan melalui SMS. Apabila akan mengirim SMS, maka cukup isikan (*insert*) saja ke dalam tabel *outbox* berupa nomor tujuan pada *field* **DestinationNumber** dan isi SMSnya pada *field* **TextDecoded**. SMS yang telah dikirimkan, akan dipindahkan ke tabel *sentitems*, sehingga setelah terkirim tabel *outbox* akan selalu kosong. Isi dari SMS yang dikirimkan adalah jenis kendaraan yang akan dipakai oleh karyawan.

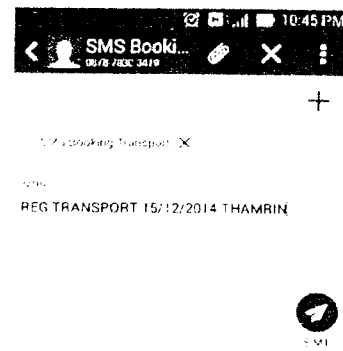
Sistem akan melakukan pengecekan jumlah mobil kantor yang masih tersedia dan lokasi tujuan, apabila mobil kantor masih tersedia program akan memberikan perintah untuk mengirimkan isi pesan yaitu "TRANSPORT TANGGAL(DD/MM/YYYY) : (nomor) menggunakan mobil kantor. Terima kasih". Jika semua mobil kantor sudah *full booked* maka program akan memberikan perintah untuk mengirimkan isi pesan yaitu "TRANSPORT TANGGAL(DD/MM/YYYY) : (nomor) menggunakan *voucher* taksi, silahkan ambil *voucher* di receptionist

4. PENGUJIAN DAN ANALISA DATA

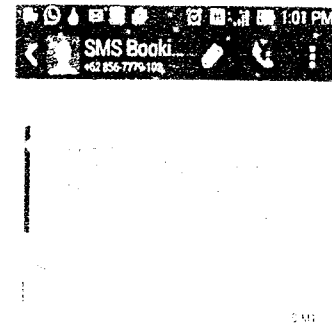
4.1 Pengujian Sistem

Aplikasi SMS *Gateway* dengan *software* Gammu pada sistem transportasi *booking* ditujukan untuk memudahkan karyawan dalam memesan kendaraan saat pergi ke klien. Keberhasilan sistem ini ditandai dengan berjalannya aplikasi SMS *Gateway* dan nilai parameter yang diuji seperti *throughput*, *delay*, dan *packet loss* sistem memenuhi standar ITU-T G11.4.

Sistem transportasi *booking* berbasis SMS *Gateway* dengan *software* Gammu sebelumnya harus terhubung dengan modem Huawei E153. Karyawan akan mengirimkan SMS ke server SMS *Gateway* kemudian server secara otomatis akan mengirimkan SMS konfirmasi kembali kepada karyawan. Keberhasilan ditandai dengan karyawan yang menerima SMS dari server SMS *Gateway* jika format SMS yang dikirim benar. Jika mobil kantor tersedia maka isi pesan SMS yang dikirimkan seperti ditunjukkan pada Gambar 4.1. Jika mobil kantor sudah *full booked* maka isi pesan SMS yang dikirimkan ditunjukkan pada Gambar 4.2, dan selanjutnya karyawan sebelum pergi ke klien harus mengambil *voucher* taksi terlebih dahulu di receptionist.



(a)

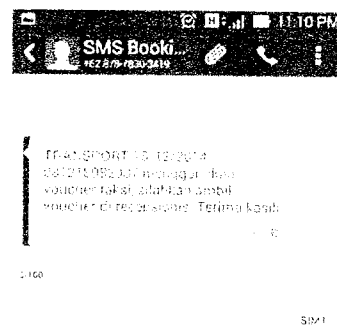


(b)

Gambar 4.1 Format Benar (Mobil): (a) SMS Dikirim (b) SMS Diterima



(a)



(b)

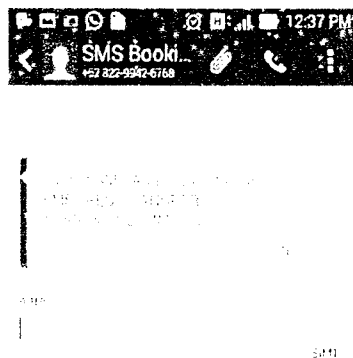
Gambar 4.2 Format Benar (Taksi): (a) SMS Dikirim (b) SMS Diterima

Selain itu keberhasilan juga ditandai dengan karyawan yang menerima SMS dari server SMS *Gateway* jika format SMS yang dikirim salah, seperti ditunjukkan oleh Gambar 4.3. Jika demikian

maka karyawan harus mengirimkan SMS kembali dengan format yang benar.



(a)



(b)

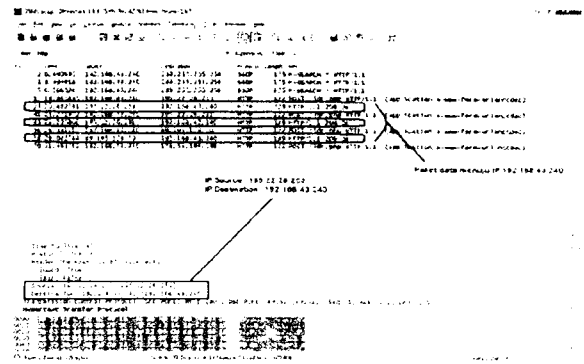
Gambar 4.3 Format Salah : (a) SMS Dikirim (b) SMS Diterima

4.2 Analisa Data

Setelah pengujian sistem dilakukan maka proses berikutnya adalah penentuan QoS sistem. Dari hasil pengujian sistem secara keseluruhan maka parameter QoS sistem yang dalam hal ini yaitu *throughput*, *delay* dan *packet loss* dapat diketahui. Pengukuran nilai *throughput*, *delay* dan *packet loss* dilakukan dengan menggunakan *software* Wireshark. Adapun analisa pengukuran parameter-parameter tersebut adalah sebagai berikut :

4.2.1 Analisa Pengukuran Throughput

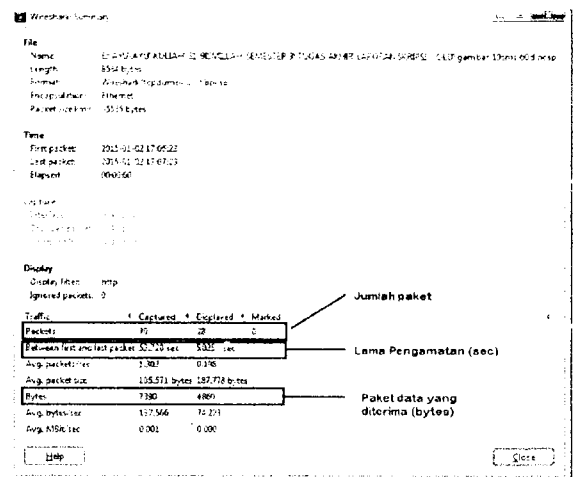
Throughput adalah kecepatan (*rate*) data efektif yang diukur dalam *bps*. *Throughput* merupakan jumlah total kedatangan paket yang sukses diamati pada destinasi selama interval waktu tertentu. Pengukuran *throughput* pada tugas akhir ini dilakukan dengan cara pengamatan saat pengiriman SMS dari sisi pengirim dalam hal ini karyawan DataOn (IP 195.22.26.253) ke sisi penerima yaitu server (IP 192.168.43.240) dengan menggunakan *software* Wireshark seperti diperlihatkan pada Gambar 4.4.



Gambar 4.4 IP Source dan IP Destination

Informasi SMS yang dikirimkan oleh karyawan dikonversi menjadi data yang akan ditransmisikan melalui *network* atau jaringan. Pada layer aplikasi, informasi SMS tersebut sudah dikonversi menjadi data. Setelah data diformat dan dikelompokkan sehingga tidak akan tercampur dengan data aplikasi lainnya, *transport layer* akan memecah data tersebut menjadi bagian-bagian yang disebut *segment*. Pada tahap ini, data diatur sedemikian rupa agar tidak sampai hilang di tengah jalan dengan menggunakan *protocol* TCP, sehingga data yang hilang bisa dikirimkan kembali. Kemudian tiap *segment* yang dibentuk pada *transport layer* tadi diubah lagi menjadi paket-paket.

Analisa *throughput* dalam tugas akhir ini diperoleh dengan cara membandingkan jumlah paket yang diterima oleh server. Hasil *capture throughput* dengan *software* Wireshark dapat dilihat pada submenu *Statistic Summary*, seperti ditunjukkan oleh Gambar 4.5.



Gambar 4.5 Capture Throughput dengan Wireshark

Setelah dilakukan beberapa proses pengambilan data melalui *capture* Wireshark maka didapatkan hasil akhir sesuai Tabel 4.1. Ukuran paket yang diterima dari hasil pengukuran Wireshark dalam satuan *bytes* sehingga sebelum menghitung nilai *throughput* harus dikonversi

terlebih dahulu ke dalam satuan *bit*, dimana 1 *byte* = 8 *bit*.

Paket data yang diterima = $4860 \times 8 = 38880$ *bit*

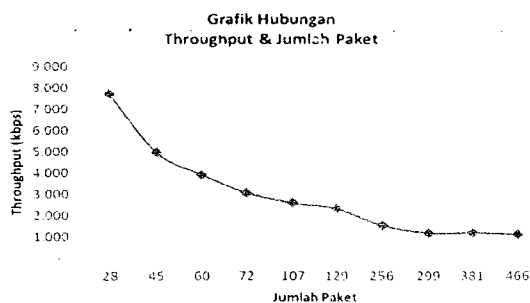
Dengan menggunakan Persamaan 2.1 didapat nilai *throughput*, sebagai berikut :

$$\begin{aligned} \text{Throughput} &= \frac{38880 \text{ bit}}{5,035 \text{ sec}} \\ &= 7721,946 \text{ bps} \\ &= 7,722 \text{ kbps} \end{aligned}$$

Tabel 4.1 Hasil Perhitungan *Throughput*

No	Jumlah Paket	Jumlah Byte	Between First & Last Packet (s)	Throughput (kbps)
1	28	4860	5,035	7,722
2	45	5091	8,192	4,972
3	60	5349	10,915	3,920
4	72	5706	14,874	3,069
5	107	7203	22,269	2,583
6	129	8213	28,413	2,312
7	256	11161	58,564	1,525
8	299	12457	85,945	1,160
9	381	17115	115,764	1,183
10	466	20195	147,353	1,096
Rata-rata	173,6	7715,5	49,732	2,955

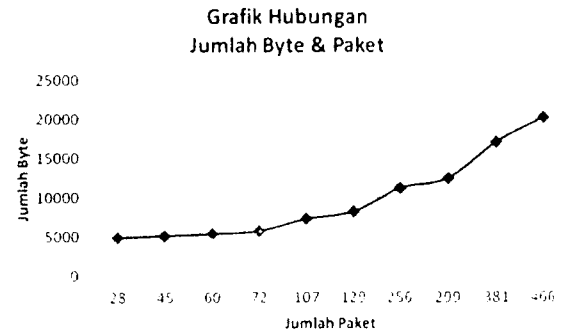
Berdasarkan hasil analisis dari Tabel 4.1 didapat nilai *throughput* rata-rata sebesar 2,955 kbps. Terlihat nilai *throughput* mengalami penurunan ketika ukuran paket (jumlah *byte*) diperbesar. Hal ini disebabkan karena ukuran paket yang dikirim semakin besar sedangkan waktu simulasinya sama yaitu 60 detik. Nilai *overhead* pada setiap ukuran paket dapat mempengaruhi besar *throughput*, dimana ketika ukuran paket semakin besar maka jumlah paket yang diterima mendekati setengah dari ukuran paket yang lebih kecil dengan *overhead* yang sama. Gambar 4.6 menunjukkan grafik hubungan *throughput* dan jumlah paket.



Gambar 4.6 Grafik Hubungan *Throughput* dan Jumlah Paket

Throughput sifatnya adalah dinamis tergantung trafik yang sedang terjadi. Sedangkan hubungan antara jumlah paket berbanding lurus dengan ukuran paket atau jumlah *byte*, semakin banyak jumlah paket yang diterima maka ukuran

paket tersebut juga semakin besar. Grafik hubungan jumlah *byte* dengan jumlah paket diperlihatkan pada Gambar 4.7.



Gambar 4.7 Grafik Hubungan Jumlah Byte dan Jumlah Paket

4.2.2 Analisa Pengukuran *Delay*

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Untuk menghitung *delay* dari sistem ini, maka dibutuhkan waktu pengiriman pada sisi *transmitter* (server) dan waktu penerimaan data di sisi *receiver* (*handphone*).

Delay data diukur saat server menerima maupun mengirimkan SMS notifikasi ke nomor *handphone* karyawan yang telah terdaftar dalam *database* DataOn Internal System. Berdasarkan hasil dari *capture* data yang didapat melalui *software* Wireshark dan perhitungan dengan Persamaan 2.2 maka dapat diketahui besar *delay* dari komunikasi antara server ke *handphone* karyawan. Hasilnya dapat dilihat pada Tabel 4.2 sebagai berikut.

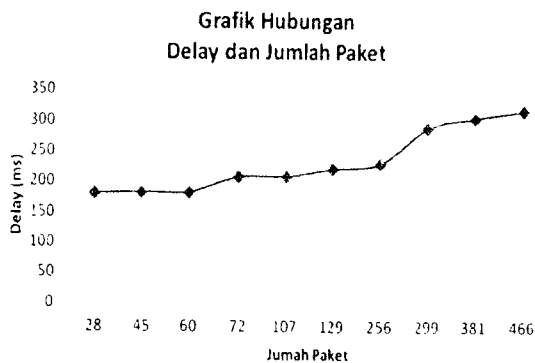
$$\begin{aligned} \text{Delay} &= \frac{5,035}{28} \text{ s} \\ &= 0,180 \text{ s} \\ &= 180 \text{ ms} \end{aligned}$$

Tabel 4.2 Hasil Perhitungan *Delay*

No	Jumlah Paket	Between First & Last Packet (s)	Delay (ms)
1	28	5,035	180
2	45	8,192	182
3	60	10,915	182
4	72	14,874	207
5	107	22,269	208
6	129	28,413	220
7	256	58,564	229
8	299	85,945	287
9	381	115,764	304
10	466	147,353	316
Rata-rata	173,6	49,732	231,5

Berdasarkan Tabel 4.2 perbandingan *delay* berdasarkan jumlah paket terlihat semakin besar jumlah paket maka semakin besar pula nilai *delay*-nya. Hal ini disebabkan karena ketika jumlah paket lebih besar maka pengiriman paket membutuhkan

waktu yang lebih lama. Menurut standar dari ITU-T G11.4 untuk kualitas jaringan yang baik disarankan memiliki nilai *delay* sebesar 150 – 300 ms. Sehingga nilai rata-rata *delay* yaitu 231,5 ms memiliki nilai yang masih diterima untuk melakukan komunikasi SMS. Gambar 4.8 menunjukkan hubungan *delay* dengan jumlah paket.



Gambar 4.8 Grafik Hubungan *Delay* dan Jumlah Paket

4.2.3 Analisa Pengukuran *Packet Loss*

Pengukuran *packet loss* dalam simulasi ini dilakukan dengan menggunakan cara pada jumlah seluruh TCP yang diterima oleh *handphone* karyawan agar diamati dan dibandingkan dengan paket SMS yang dikirim oleh server. Data-data yang terdapat pada *capture* Wireshark untuk *packet loss* dapat dilihat melalui sub menu *Endpoints*. Dalam menghitung *packet loss* berdasarkan hasil pengukuran dari *software* Wireshark dapat digunakan Persamaan 2.3 sehingga diperoleh :

$$\text{Packet Loss} = \frac{(28 - 28)}{28} \times 100\% = 0\%$$

Berdasarkan perhitungan *packet loss* secara matematis seperti hasil *packet loss* pada saat pengiriman 1 SMS, maka hasil perhitungan *packet loss* untuk jumlah paket yang berbeda, bila dihitung dengan cara yang sama diperoleh hasil seperti pada Tabel 4.3.

Dari Tabel 4.3 diperoleh nilai rata-rata *packet loss* pada saat melakukan pengiriman SMS sebesar 0%, yang artinya paket yang dikirim dan diterima dari *handphone* karyawan menuju server atau sebaliknya tidak ada paket yang mengalami *broken* (rusak) ataupun hilang (*lost*) pada saat pengiriman ataupun penerimaan data.

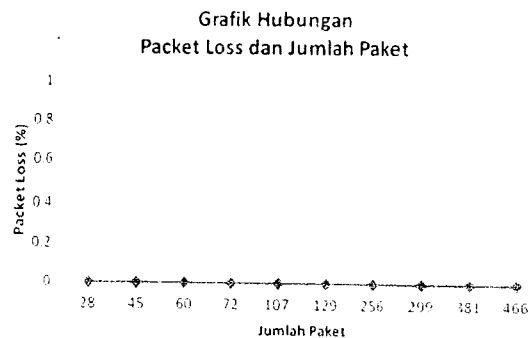
Biasanya hal yang menjadi penyebab adanya *packet loss*, pada saat *request* ataupun *receive* data adalah kegagalan pada jaringan, kepadatan trafik di jaringan, karena kesalahan *hardware*, dan keterbatasan *bandwidth* saat melakukan *transmisi* data. Jika banyak paket yang hilang, maka kualitas data yang diterima kedua

client akan menjadi buruk. Gambar 4.9 menunjukkan hubungan jumlah paket dengan *packet loss*.

Tabel 4.3 Hasil Perhitungan *Packet Loss*

No	Jumlah Paket (Tx)	Jumlah Paket (Rx)	Packet Loss (%)
1	28	28	0
2	45	45	0
3	60	60	0
4	72	72	0
5	107	107	0
6	129	129	0
7	256	256	0
8	299	299	0
9	381	381	0
10	466	466	0
Rata-rata	173,6	173,6	0

Berdasarkan rekomendasi ITU-T G11.4 bahwa nilai *packet loss* yang masih bisa di toleransi adalah 5%. Dari hasil pengukuran terlihat nilai *packet loss* masih termasuk dalam rekomendasi, sehingga menunjukkan jaringan ini memiliki performansi yang sangat baik. Hal ini dikarenakan protokol yang digunakan adalah TCP yang memiliki kemampuan untuk pengecekan paket data yang hilang ataupun rusak dan mengirimkannya kembali.



Gambar 4.9 Grafik Hubungan *Packet Loss* dan Jumlah Paket

5. KESIMPULAN

Berdasarkan keseluruhan uji coba dan analisa yang telah dilakukan, maka didapatkan kesimpulan akhir, yaitu :

1. Sistem transportasi booking berbasis SMS *Gateway* dapat berjalan dengan baik, keberhasilan ditandai dengan karyawan yang menerima SMS dari server SMS *Gateway* jika format SMS yang dikirim benar serta mendapat konfirmasi SMS apabila format SMS yang dikirim salah.
2. Performansi sistem transportasi booking berbasis SMS *Gateway* termasuk dalam kategori baik menurut standar ITU-T G11.4. Dilihat dari parameter QOS yang diukur yakni

throughput rata-rata sebesar 2,955 kbps, *delay* rata-rata sebesar 231,5 ms dan *packet loss* rata-rata sebesar 0%.

3. Semakin besar jumlah paket maka *delay* yang terjadi semakin besar namun nilai *throughput* yang didapat semakin kecil.

DAFTAR PUSTAKA

- [1] Zakaria, Teddy Markus dan Josef Widioadhi. 2006. *Aplikasi SMS untuk Berbagai Keperluan*. Bandung: Informatika Bandung.
- [2] Purnomo, Adi. 2007. *Pemrograman Java 2: Membangun Beragam Aplikasi Layanan SMS*. Jakarta: Salemba Infotek.
- [3] Rosidi, Romzi Imron. 2004. *Membuat sendiri SMS Gateway (GSMF) Berbasis Protokol SMPP*. Yogyakarta: Andi.
- [4] Wahana Komputer. 2005. *Pengembangan Aplikasi Sistem Informasi Akademik Berbasis SMS dengan Java*. Jakarta: Salemba.
- [5] Endi Dwi Kristianto. 2012. Menghitung Delay Paket Jaringan Menggunakan Wireshark. <http://teknologi.kompasiana.com/terapan/2012/10/18/menghitung-delay-paket-jaringan-menggunakan-wireshark-502650.html> diakses 13 Desember 2014
- [6] David Sudana. 2009. Menenal Cara Kerja SMS Gateway. <https://duniadhana.wordpress.com/2009/08/02/mengenal-cara-kerja-sms-gateway> diakses 14 Desember 2014.
- [7] Dwi Herry Jayanto. 2006. Analisa Performansi Media Access Control pada Jaringan Ethernet. <http://dherryjayanto.blogspot.com/2006/12/analisa-performansi-media-access.html> diakses 5 Januari 2015.
- [8] Niko Rahmadi Wiharto. 2014. Proses yang Terjadi Pada Saat Pengiriman Data Berdasarkan 7 OSI Layer. <http://www.pintarkomputer.com/2014/09/proses-yang-terjadi-pada-saat-pengiriman-data-berdasarkan-7-osi-layer.html> diakses 5 Januari 2015
- [9] Rosihan Ari. 2009. Setting Gammu untuk Aplikasi SMS Gateway. <http://blog.rosihanari.net/setting-gammu-untuk-aplikasi-sms-gateway/> diakses 5 Januari 2015
- [10] International Telecommunication Union. 2003. Standard G.114. <http://www.itu.int/ITU-T/publications> diakses 31 Januari 2015