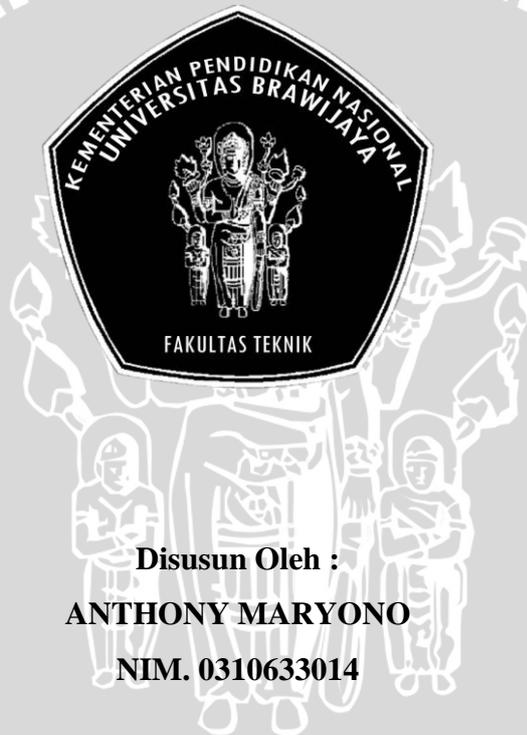


**Perancangan Sistem Monitoring Koneksi *Router* di
Fakultas Teknik Universitas Brawijaya Menggunakan
SMS Gateway**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

UNIVERSITAS BRAWIJAYA



Disusun Oleh :

ANTHONY MARYONO

NIM. 0310633014

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS BRAWIJAYA

MALANG

2010

**Perancangan Sistem Monitoring Koneksi *Router* di
Fakultas Teknik Universitas Brawijaya Menggunakan
*SMS Gateway***

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

UNIVERSITAS BRAWIJAYA



Disusun Oleh :
ANTHONY MARYONO
NIM. 0310633014

Telah Diperiksa dan Disetujui Oleh:
Dosen Pembimbing

Rudy Yuwono, ST., M.Sc
19710615 199802 1 003

Ir. Endah Budi P., MT
19621116 198903 2 002

**Perancangan Sistem Monitoring Koneksi Router di
Fakultas Teknik Universitas Brawijaya Menggunakan
SMS Gateway**

SKRIPSI

Disusun Oleh :

ANTHONY MARYONO

NIM. 0310633014

Skripsi ini telah diuji dan dinyatakan lulus
pada tanggal 4 Agustus 2010

Majelis Penguji

Ir. Sholeh Hadi Pramono, MS
NIP. 19580728 198701 1 001

Goegoes Dwi Nusantoro, ST., MT
NIP. 19711013 200604 1 001

Dwi Fadilla Kurniawan, ST., MT
NIP. 19720630 200003 1 002

Mengetahui,
Ketua Jurusan Teknik Elektro

Rudy Yuwono, ST., M.Sc
NIP. 19710615 199802 1 003

KATA PENGANTAR

Puji Syukur kehadiran Tuhan YME karena dengan berkat rahmat dan karunia-Nya penyusunan skripsi dengan judul “Perancangan Sistem Monitoring Koneksi *Router* di Fakultas Teknik Universitas Brawijaya Menggunakan SMS *Gateway*” dapat diselesaikan. Penulis menyadari bahwa kajian ini tidak akan mencapai titik akhir penyelesaian tanpa bantuan berbagai pihak, karenanya penulis mengucapkan terima kasih kepada:

1. Mama dan alm. Papa tercinta atas segala jerih payah, cinta, kasih sayang, dan kesabarannya dalam memberikan dukungan doa, semangat, dan materiil kepada Ananda.
2. Bapak Rudy Yuwono, ST., M.Sc selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
3. Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
4. Bapak Ali Mustofa, ST., MT selaku KKDK Teknik Telekomunikasi, Jurusan Teknik Elektro.
5. Bapak Rudy Yuwono, ST., M.Sc dan Ibu Ir. Endah Budi P., MT selaku pembimbing I dan pembimbing II yang telah memberikan banyak bimbingan, ide, motivasi, pengarahan, saran dan masukan selama pengerjaan skripsi ini.
6. Bapak dan Ibu Dosen serta Karyawan Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
7. Stefanny Damayanti Rihiga yang dengan setia terus memberikan semangat dan dorongan selama pengerjaan skripsi ini.
8. Teman – teman “SilverGen” Elektro 2003, Tim Kerja Core System, serta semua pihak yang telah banyak membantu penulis selama ini.

Tiada yang sempurna di dunia ini, tersadar bahwa skripsi ini sangat jauh dari kesempurnaan. Karenanya, segala kritik dan saran yang sifatnya membangun dari pembaca tentang isi skripsi ini akan diterima dengan senang hati. Akhir kata, penulis berharap, semoga skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Malang, 30 Juli 2010

Penyusun

DAFTAR ISI

| | |
|---|------------|
| KATA PENGANTAR | iii |
| DAFTAR ISI | iv |
| DAFTAR GAMBAR | vii |
| DAFTAR TABEL | ix |
| DAFTAR SINGKATAN | x |
| ABSTRAK | xi |
| | |
| BAB I PENDAHULUAN | |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Ruang Lingkup | 2 |
| 1.4 Tujuan | 3 |
| 1.5 Sistematika Penulisan | 3 |
| | |
| BAB II TINJAUAN PUSTAKA | |
| 2.1 <i>Transmission Control Protocol/Internet Protocol</i> | 4 |
| 2.1.1 Model Referensi OSI (<i>Open System Interconnection</i>)... 4 | |
| <i>Physical Layer</i> | 5 |
| <i>Data Link Layer</i> | 6 |
| <i>Network Layer</i> | 7 |
| <i>Transport Layer</i> | 7 |
| <i>Session Layer</i> | 8 |
| <i>Presentation Layer</i> | 8 |
| <i>Application Layer</i> | 9 |
| 2.1.2 <i>Transmission Control Protocol/Internet Protocol</i> 9 | |
| <i>Network Interface Layer</i> | 11 |
| <i>Internet Layer</i> | 11 |
| <i>Internet Protocol</i> | 11 |
| <i>Transport Layer</i> | 12 |
| TCP (<i>Transmission Control Protocol</i>). 12 | |



| | | |
|-------|---|----|
| | UDP (<i>User Datagram Protocol</i>)..... | 14 |
| | <i>Application Layer</i> | 15 |
| 2.2 | Router..... | 15 |
| 2.3 | SMS Gateway..... | 17 |
| 2.4 | Telepon Seluler..... | 18 |
| 2.4.1 | Sony Ericsson..... | 18 |
| | Pin out Telepon Seluler Sony Ericsson W830i. | 19 |
| 2.5 | SHORT MESSAGE SERVICES (SMS) | 19 |
| 2.5.1 | Umum | 19 |
| 2.5.2 | Konsep Dasar Pengiriman SMS..... | 21 |
| 2.5.3 | Format Data SMS..... | 22 |
| 2.5.4 | Prinsip Kerja SMS..... | 22 |
| | PDU untuk Kirim SMS ke <i>SMS-Centre</i> | 23 |
| | PDU untuk SMS Terima dari <i>SMS-Centre</i> | 28 |
| 2.6 | AT Command..... | 28 |
| 2.7 | Performansi SMS..... | 29 |
| 2.7.1 | Delay End to End..... | 29 |
| 2.7.2 | Delay Transmisi..... | 30 |
| 2.7.3 | Delay Propagasi..... | 31 |
| 2.7.4 | Delay Antrian..... | 31 |

BAB III METODOLOGI

| | | |
|-------|---|----|
| 3.1 | Jenis Data dan Cara Mendapatkan Data..... | 33 |
| 3.1.1 | Data Primer | 33 |
| 3.1.2 | Data Sekunder | 33 |
| 3.2 | Variabel dan Cara Analisis Data yang Digunakan..... | 34 |
| 3.2.1 | Variabel <i>Delay</i> Transmisi..... | 34 |
| 3.2.2 | Variabel <i>Delay</i> Propagasi..... | 34 |
| 3.2.3 | Variabel <i>Delay</i> Antrian..... | 35 |
| 3.3 | Rangka Solusi Permasalahan..... | 36 |



BAB IV PERANCANGAN SISTEM

| | |
|---|----|
| 4.1 Perancangan Sistem | 40 |
| 4.1.1 Perangkat Keras Sistem..... | 42 |
| 4.1.2 Perangkat Lunak..... | 42 |
| 4.1.3 Prinsip Kerja Sistem..... | 42 |
| 4.1.4 Rangkaian <i>Interface</i> DCU-60 | 44 |

BAB V PENGUJIAN DAN ANALISIS SISTEM

| | |
|--|----|
| 5.1 Perhitungan <i>Delay</i> | 46 |
| <i>Delay End to End</i> | 46 |
| 5.1.1 Perhitungan <i>Delay</i> Transmisi..... | 46 |
| 5.1.2 Perhitungan <i>Delay</i> Propagasi..... | 47 |
| 5.1.3 Perhitungan <i>Delay</i> Antrian..... | 49 |
| 5.2 Pengaturan <i>Baudrate</i> | 51 |
| 5.3 Pengujian Sistem..... | 51 |
| 5.3.1 Pengujian <i>AT Command</i> dan Format Data SMS..... | 51 |
| 5.3.2 Pengujian <i>Software Kannel</i> | 54 |
| 5.3.3 Pengujian Pengiriman SMS Menggunakan <i>Kannel</i> .. | 56 |
| 5.3.4 Pengujian <i>Script</i> Monitoring Koneksi <i>Router</i> | 60 |
| 5.3.5 Pengujian Keseluruhan Sistem..... | 61 |
| 5.4.6 Pengujian Waktu Tunda Pengiriman SMS..... | 62 |

BAB VI PENUTUP

| | |
|----------------------|----|
| 6.1 Kesimpulan | 66 |
| 6.2 Saran | 66 |

| | |
|-----------------------------|----|
| DAFTAR PUSTAKA | 68 |
|-----------------------------|----|

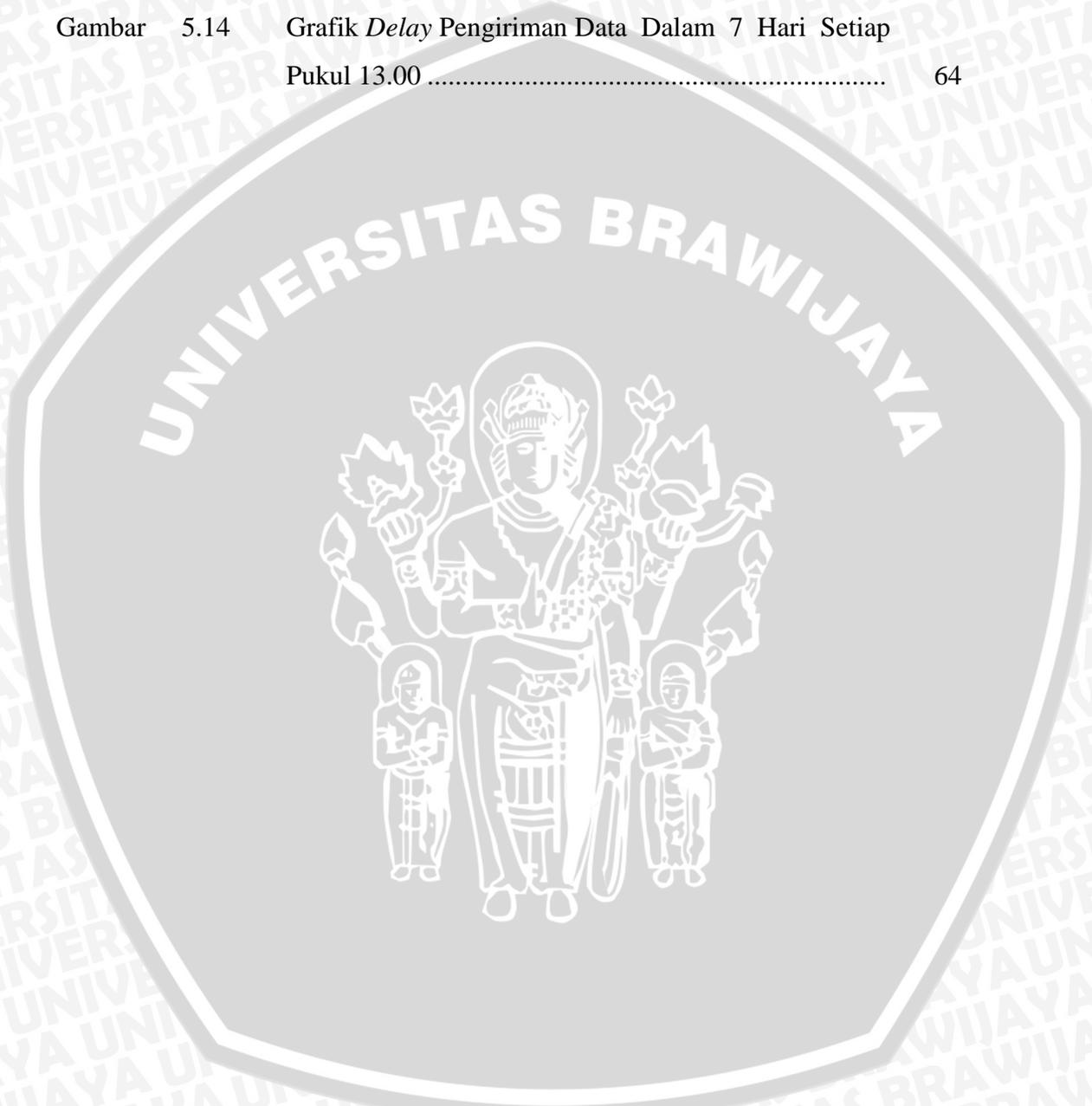
LAMPIRAN



DAFTAR GAMBAR

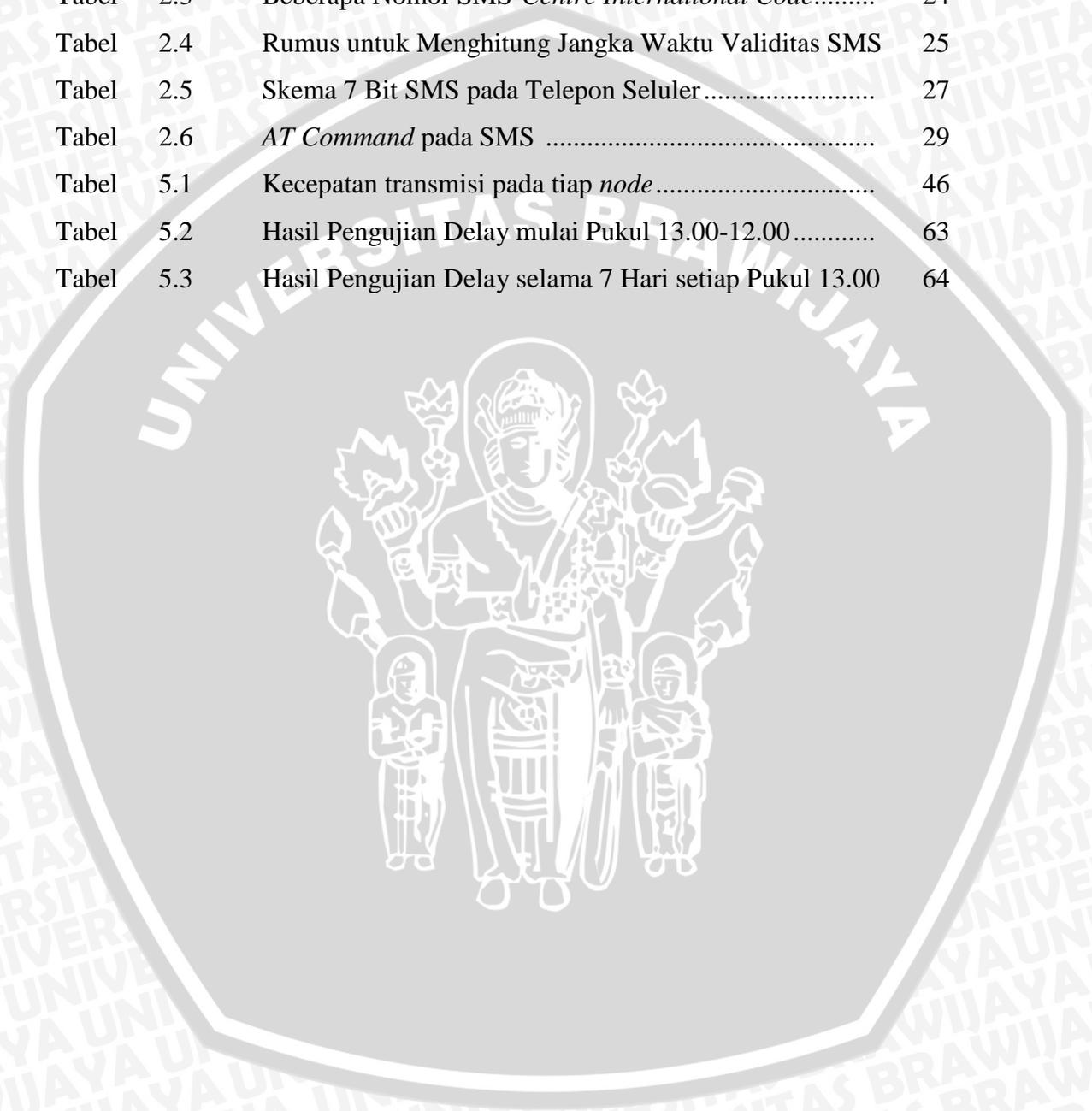
| | | | |
|--------|------|---|----|
| Gambar | 2.1 | Model OSI (<i>Open System Interconnection</i>) | 4 |
| Gambar | 2.2 | Susunan <i>Protocol</i> TCP/IP dan Model OSI..... | 9 |
| Gambar | 2.3 | <i>Layer</i> TCP/IP(<i>Transmission Control Protocol/Internet Protocol</i>) | 10 |
| Gambar | 2.4 | <i>Header</i> TCP (<i>Transmission Control Protocol</i>)..... | 12 |
| Gambar | 2.5 | Mekanisme <i>Three Way Handshake</i> | 14 |
| Gambar | 2.6 | <i>Header</i> UDP | 15 |
| Gambar | 2.7 | Susunan <i>Pinout</i> Sony Ericsson W830i..... | 19 |
| Gambar | 2.8 | Bentuk Jaringan SMS..... | 20 |
| Gambar | 2.9 | Skema Konsep Dasar Pengiriman SMS | 21 |
| Gambar | 2.10 | <i>Delay End to End</i> pada Jaringan SMS | 30 |
| Gambar | 3.1 | Penghitungan <i>Delay</i> Transmisi | 34 |
| Gambar | 3.2 | Penghitungan <i>Delay</i> Propagasi | 35 |
| Gambar | 3.3 | Penghitungan <i>Delay</i> Antrian..... | 36 |
| Gambar | 3.4 | Diagram Alir Solusi Permasalahan | 37 |
| Gambar | 3.5 | Diagram blok sistem | 38 |
| Gambar | 4.1 | Perancangan Sistem | 40 |
| Gambar | 4.2 | Topologi Jaringan <i>Intranet</i> FT Universitas Brawijaya. | 41 |
| Gambar | 4.2 | Diagram Alir SMS Gateway Kannel..... | 43 |
| Gambar | 4.3 | Diagram Alir Program Sistem Monitoring Koneksi Server | 44 |
| Gambar | 4.4 | Antarmuka HP dengan Port USB pada PC | 45 |
| Gambar | 5.1 | <i>Delay End to End</i> pada Jaringan GSM | 46 |
| Gambar | 5.2 | Blok Diagram Pengujian Telepon Seluler..... | 52 |
| Gambar | 5.3 | Hasil Pengujian <i>AT Command</i> | 53 |
| Gambar | 5.4 | Blok Diagram Pengujian Software Kannel | 55 |
| Gambar | 5.5 | Hasil Pengujian Bearerbox..... | 55 |
| Gambar | 5.6 | Hasil Pengujian SMSbox | 56 |
| Gambar | 5.7 | Blok Diagram Pengujian Pengiriman SMS..... | 57 |
| Gambar | 5.8 | Hasil <i>Report</i> dari Pengiriman SMS..... | 57 |

| | | |
|-------------|---|----|
| Gambar 5.9 | Pengecekan Status Pengiriman SMS | 58 |
| Gambar 5.10 | Status Pengiriman SMS pada Bearerbox | 58 |
| Gambar 5.11 | Hasil Pengujian <i>Script</i> pod.sh | 60 |
| Gambar 5.12 | Blok Diagram Pengujian Keseluruhan..... | 61 |
| Gambar 5.13 | Grafik <i>Delay</i> Pengiriman Data dari Pukul 13.00-12.00 | 63 |
| Gambar 5.14 | Grafik <i>Delay</i> Pengiriman Data Dalam 7 Hari Setiap Pukul 13.00 | 64 |



DAFTAR TABEL

| | | | |
|-------|-----|--|----|
| Tabel | 2.1 | Pin out Sony Ericsson W830i | 19 |
| Tabel | 2.2 | Beberapa Nomor SMS-Centre National Code..... | 23 |
| Tabel | 2.3 | Beberapa Nomor SMS-Centre International Code..... | 24 |
| Tabel | 2.4 | Rumus untuk Menghitung Jangka Waktu Validitas SMS | 25 |
| Tabel | 2.5 | Skema 7 Bit SMS pada Telepon Seluler | 27 |
| Tabel | 2.6 | AT Command pada SMS | 29 |
| Tabel | 5.1 | Kecepatan transmisi pada tiap node | 46 |
| Tabel | 5.2 | Hasil Pengujian Delay mulai Pukul 13.00-12.00..... | 63 |
| Tabel | 5.3 | Hasil Pengujian Delay selama 7 Hari setiap Pukul 13.00 | 64 |



DAFTAR SINGKATAN



| | |
|---------------|--|
| BSC | : <i>Base Station Controller</i> |
| BTS | : <i>Base Transceiver Stations</i> |
| CDMA | : <i>Code Division Multiple Access</i> |
| DNS | : <i>Domain Name System</i> |
| ESME | : <i>External Messaging Entities</i> |
| ETSI | : <i>European Telecommunication Standart Institute</i> |
| FTP | : <i>File Transfer Protocol</i> |
| GSM | : <i>Global System for Mobile Communication</i> |
| HLR | : <i>Home Location Register</i> |
| IP | : <i>Internet Protocol</i> |
| MS | : <i>Mobile Station</i> |
| MSC | : <i>Message Station Controller</i> |
| NNTP | : <i>Network News Transfer Protocol</i> |
| OSI | : <i>Open System Interconnection</i> |
| PC | : <i>Personal Computer</i> |
| PDU | : <i>Protocol Data Unit</i> |
| SMS | : <i>Short Message Services</i> |
| SMSC | : <i>Short Message Services Centre</i> |
| SMTP | : <i>Simple Mail Transfer Protocol</i> |
| STP | : <i>Signal Transfer Point</i> |
| TCP | : <i>Transmission Control Protocol</i> |
| TDMA | : <i>Time Division Multiple Access</i> |
| TELNET | : <i>Telecommunication Network</i> |
| UDP | : <i>User Datagram Protocol</i> |
| USB | : <i>Universal Serial Bus</i> |
| VLR | : <i>Visitor Location Register</i> |

ABSTRAK

ANTHONY MARYONO, 2010. Perancangan Sistem Monitoring Koneksi *Router* di Fakultas Teknik Universitas Brawijaya Menggunakan SMS *Gateway*. Skripsi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.

Dosen Pembimbing : Rudy Yuwono, ST., M.Sc. dan Ir. Endah Budi P.,MT.

Perkembangan jaringan *internet* telah menjadi *trend* yang diikuti oleh kalangan akademis tetapi juga oleh kalangan umum karena *internet* menyediakan akses untuk layanan telekomunikasi dan sumber daya informasi untuk jutaan pemakainya yang tersebar di seluruh dunia. Kemudahan dalam mendapatkan informasi semakin dibutuhkan dalam kegiatan-kegiatan manusia, baik oleh perorangan atau suatu perusahaan dalam menjalankan kegiatannya. Jaringan yang membentuk *internet* bekerja berdasarkan suatu set protokol standar yang digunakan untuk menghubungkan jaringan komputer dan mengamati lalu lintas dalam jaringan. Untuk terhubung ke dalam jaringan *internet* setiap jaringan komputer memiliki *router* yang bertugas dalam pelayanan jaringan, mengatur lalu lintas data dalam sebuah jaringan, dan menyediakan *resource* yang dapat dipakai oleh komputer lain yang terhubung dalam jaringannya. Sehingga apabila koneksi jaringan *internet* pada *router* terputus, maka dapat dipastikan komputer lain yang terhubung dalam jaringannya ikut terputus juga.

Pada skripsi ini membahas perancangan sistem monitoring koneksi *router* dengan menggunakan SMS *gateway* kannel. Analisis performansi dititikberatkan pada parameter SMS yaitu *Delay End to End*. Pada perhitungan didapatkan nilai *Delay End to End* sebesar 0,02724 detik, sedangkan pada pengujian pengiriman data sebanyak 24 kali mulai pukul 13.00-12.00 didapat waktu rata-rata sebesar 8,2 detik, dan pada pengujian pengiriman data selama 7 hari setiap pukul 13.00 didapat waktu rata-rata sebesar 9 detik. Performansi pengiriman SMS paling baik pada pukul 23.00-06.00 dengan *Delay End to End* sebesar 6,7-7,7 detik.

Kata Kunci : *Internet, Router, Koneksi, SMS, Delay End to End.*

BAB I PENDAHULUAN

1.1 Latar belakang

Router merupakan komputer yang bertugas dalam pelayanan jaringan, mengatur lalu lintas data dalam sebuah jaringan, dan menyediakan *resource* yang dapat dipakai oleh komputer lain yang terhubung dalam jaringannya. Sehingga apabila koneksi jaringan intranet pada *router* terputus, maka dapat dipastikan komputer lain yang terhubung dalam jaringannya ikut terputus juga. Kinerja administrator (admin) *router* sangatlah vital dalam memonitoring jaringan agar berjalan sebagaimana mestinya. Sehingga dengan adanya sistem monitoring koneksi *router* yang bersifat *mobile* dapat memudahkan kinerja admin *router* dalam memonitoring jaringan.

Kemudahan dalam mendapatkan informasi semakin dibutuhkan dalam kegiatan-kegiatan manusia, baik oleh perorangan atau suatu perusahaan dalam menjalankan kegiatannya. Untuk menunjang kebutuhan tersebut diperlukan inovasi-inovasi dalam menyediakan layanan informasi dengan cepat dan mudah. Dengan adanya *SMS Gateway* yang memungkinkan kita mengirim SMS melalui jaringan TCP/IP. *Short Messaging Service* (SMS) merupakan salah satu fitur dari GSM yang dikembangkan dan distandarisasi oleh *European Telecommunication Standart Institute* (ETSI). SMS merupakan salah satu media yang banyak digunakan oleh masyarakat sekarang ini, karena SMS memiliki tarif yang sangat murah dibandingkan berbicara langsung dengan nomor yang dituju.

SMS Gateway merupakan pintu gerbang bagi penyebaran Informasi dengan menggunakan SMS. Selain itu dengan adanya *SMS Gateway* anda dapat mengotomatisasi pesan-pesan yang ingin dikirim. Untuk kebutuhan *SMS Gateway* tidak terlalu berlebihan dan juga fleksibel karena bisa dibuat dengan PC maupun Notebook. Yang mutlak dibutuhkan adalah sebuah komputer, sebuah ponsel dengan kabel data (Kabel berantarmuka serial yang dapat menghubungi ponsel dengan PC) atau dapat menggunakan *InfraRed* dan piranti lunak sebagai *SMS Gateway*.

Skripsi ini akan membahas tentang bagaimana membuat suatu sistem monitoring *router* dengan menggunakan *SMS Gateway*.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang dijelaskan diatas, maka rumusan masalah dapat ditekankan pada:

1. Bagaimana konfigurasi program SMS *Gateway* Kannel untuk menunjang monitoring disetiap *router* yang ada di Fakultas Teknik Universitas Brawijaya.
2. Bagaimana pengujian dan analisa hasil perencanaan SMS *Gateway* untuk monitoring koneksi disetiap *router* yang ada di Fakultas Teknik Universitas Brawijaya.
3. Bagaimana pengujian dan analisa *Delay End to End* dan transmisi data dari PC (*Personal Computer*) ke HP (*Handphone*) administrator (admin).

1.3 Ruang Lingkup

Berdasarkan rumusan masalah yang dijelaskan diatas, maka penulisan skripsi ini dibatasi pada hal-hal sebagai berikut:

1. Perancangan SMS *Gateway* untuk monitoring koneksi *Router*.
2. Dalam perancangan SMS *Gateway* digunakan SMS *Gateway* Kannel.
3. Menggunakan kabel data Sony Ericsson W830i untuk menghubungkan *handphone* ke PC.
4. *Operating System* yang digunakan adalah LINUX Centos 5.3.
5. Menggunakan jaringan GSM (*Global System for Mobile Communications*).
6. Operator telekomunikasi yang digunakan adalah Indosat (IM3) dan Telkomsel (Kartu As).
7. Aspek yang dibahas meliputi performansi jaringan SMS dalam hal ini *Delay End to End* dan transmisi data dari PC (*Personal Computer*) ke HP (*Handphone*) admin.

1.4 Tujuan

Tujuan dari penulisan skripsi ini adalah mengkaji kinerja sistem monitoring koneksi disetiap *router* di Fakultas Teknik Universitas Brawijaya menggunakan SMS *Gateway*.

1.5 Sistematika Penulisan

Penulisan skripsi ini terbagi menjadi enam bab, dengan garis besar penulisan sebagai berikut:

Bab I Pendahuluan

Pada bab ini akan menjelaskan tentang latar belakang, rumusan masalah, ruang lingkup, tujuan, metodologi dan sistematika penulisan.

Bab II Tinjauan Pustaka

Merupakan analisis materi yang digunakan sebagai dasar dalam pembahasan laporan yang dibuat.

Bab III Metodologi

Merupakan kerangka acuan di dalam memfokuskan proses pengerjaan skripsi, sehingga diperoleh hasil analisis yang sesuai dengan pokok permasalahan yang dibahas

Bab IV Perancangan Sistem

Membahas perancangan sistem monitoring koneksi *router* menggunakan SMS *gateway* dimana performansi sistem tersebut adalah *delay end to end* yang terdiri dari *delay* transmisi, *delay* propagasi, dan *delay* antrian.

Bab V Pengujian dan Analisis Sistem

Membahas pengujian dan analisis sistem monitoring koneksi *router* menggunakan SMS *gateway* dimana performansi sistem tersebut adalah *delay end to end* yang terdiri dari *delay* transmisi, *delay* propagasi, dan *delay* antrian.

Bab VI Penutup

Berisi kesimpulan dan saran yang merupakan kontribusi pokok dari hasil-hasil analisis yang dilakukan pada bab-bab sebelumnya.

BAB II

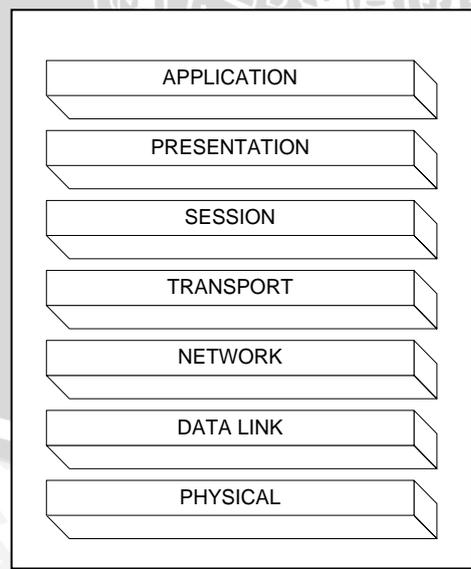
TINJAUAN PUSTAKA

2.1 TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL (TCP/IP)

2.1.1 Model Referensi OSI (*Open System Interconnection*)

Sebuah badan multinasional yang didirikan tahun 1947, yang bernama *International Standards Organization (ISO)*, adalah badan yang mengembangkan model referensi OSI yang berfungsi sebagai pemandu untuk mendefinisikan sekumpulan protokol. Model referensi OSI tetap menjadi standar yang paling umum untuk menggambarkan dan membandingkan rangkaian protokol.

OSI (*Open System Interconnection*) yang merupakan himpunan protokol berlapis, yang memungkinkan terhubungnya dua sistem yang berbeda, yang berasal dari *underlying architecture* yang berbeda pula. Jadi, tujuan dikembangkannya OSI ini adalah untuk memfasilitasi bagaimana suatu komunikasi dapat terjalin dari sistem yang berbeda tanpa memerlukan perubahan yang signifikan pada *hardware* dan *software* di tingkat *underlying*. Pada Gambar 2.1 diperlihatkan lapisan model OSI.



Gambar 2.1 Model OSI (*Open System Interconnection*)
[Sumber : www.prasimax.com]

Pendekatan untuk merancang sistem komunikasi ini dikenal sebagai arsitektur berlapis. Setiap lapisan mempunyai tanggung jawab dan peraturan khusus untuk melaksanakan tanggung jawab tersebut dan tidak ikut bertanggung jawab terhadap prosedur-prosedur pada lapisan lainnya (Heywood, 1999 : 27)

Model OSI disusun atas tujuh lapisan. Lapisan-lapisan tersebut diberi nomor dari dasar *stack* sampai yang paling atas (Heywood, 1999 : 30). Berikut adalah lapisan/layer Model OSI :

1. *Physical Layer*
2. *Data Link Layer*
3. *Network Layer*
4. *Transport Layer*
5. *Session Layer*
6. *Presentation Layer*
7. *Application Layer*

Physical Layer

Physical Layer berkomunikasi secara langsung dengan media komunikasi dan memiliki tanggung jawab mengirim dan menerima bit-bit. Sebuah *binary digit* atau bit adalah satuan terkecil dari informasi dalam komunikasi data.

Berbagai macam media digunakan untuk komunikasi data termasuk kabel listrik, serat optik, gelombang cahaya, radio, dan *microwave*. Lapisan ini akan mengetahui spesifikasi mekanik dan elektrik dari media transmisi serta antarmukanya. Jadi, lapisan yang berada di atasnya bebas dari proses menghantarkan bit-bit melalui media jaringan.

Berikut adalah fungsi dari *Physical Layer* :

- a. Karakteristik fisik media dan antarmuka.
- b. Representasi bit-bit. Maksudnya, lapisan fisik harus mampu menerjemahkan bit 0 atau 1, pengkodean, dan bagaimana mengganti sinyal 0 ke 1 atau sebaliknya.
- c. *Data rate* (laju data).
- d. Sinkronisasi bit.
- e. *Line configuration* (Konfigurasi saluran). Contoh : *point-to-point* atau *point-to-multipoint configuration*.

- f. Topologi fisik. Contoh : *mesh topology*, *star topology*, *ring topology* atau *bus topology*.

Data Link Layer

Data Link Layer bertanggung jawab untuk menyediakan komunikasi dari *node* ke *node* pada suatu jaringan lokal. Untuk menyediakan layanan ini, *Data Link Layer* harus melakukan dua fungsi yaitu harus menyediakan mekanisme alamat yang memungkinkan pesan-pesan untuk dikirimkan ke *node* yang benar, dan layer ini juga harus menerjemahkan pesan-pesan dari *Physical Layer*.

Ketika *Data Link Layer* menerima suatu pesan untuk ditransmisikan, *Data Link Layer* membentuk pesan tersebut menjadi *frame* data. *Frame* data disebut juga sebagai paket data. Layer ini menjamin informasi bebas *error* untuk ke lapisan di atasnya. Tanggung jawab utama *Data Link Layer* adalah sebagai berikut:

- a. *Framing*, yaitu membagi bit *stream* yang diterima dari *Network Layer* menjadi unit-unit data yang disebut *frame*.
- b. *Physical addressing*. Jika *frame-frame* didistribusikan ke sistem lain pada jaringan, maka data link akan menambahkan sebuah *header* di muka *frame* untuk mendefinisikan pengirim dan/atau penerima.
- c. *Flow control*. Jika *rate* atau laju *bit stream* berlebihan atau berkurang maka *flow control* akan melakukan tindakan yang menstabilkan laju bit.
- d. *Error control*. *Data Link Layer* menambah kehandalan lapisan fisik dengan penambahan mekanisme deteksi dan pentransmisian kembali *frame-frame* yang gagal terkirim.
- e. *Access control*. Jika dua atau lebih *device* dikoneksi dalam *link* yang sama, *Data Link Layer* perlu menentukan *device* yang mana yang harus dikendalikan pada saat tertentu.

Network Layer

Network Layer bertanggung jawab atas pengiriman paket dengan konsep *source-to-destination*. Adapun tanggung jawab spesifik *Network Layer* ini adalah:

- a. *Logical Addressing*.

Bila pada *Data Link Layer* diimplementasikan *physical addressing* untuk menangani pengalamatan/*addressing* secara lokal, maka pada *Network Layer* diimplementasikan *logical addressing* untuk menangani pengalamatan antar jaringan/*network*. Pada *Network Layer* ini *logical address* ditambahkan pada paket yang datang dari *Data Link Layer*.

b. *Routing*.

Jaringan-jaringan yang saling terhubung yang membentuk *internetwork* diperlukan metode *routing* sehingga paket dapat ditransfer dari satu *device* yang berada pada jaringan tertentu menuju *device* lain, yang berada pada jaringan yang lain.

Transport Layer

Transport Layer bertanggung jawab untuk pengiriman *source-to-destination (end-to-end)* dari jenis *message* tertentu. Tanggung jawab spesifik *Transport Layer* ini adalah:

- a. *Service-point addressing*. Komputer sering menjalankan berbagai macam program atau aplikasi yang berlainan dalam saat bersamaan. Untuk itu dengan *Transport Layer* ini tidak hanya menangani pengiriman/*delivery source-to-destination* dari komputer yang satu ke komputer yang lain saja namun lebih spesifik kepada *delivery* jenis *message* untuk aplikasi yang berlainan. Setiap *message* yang berlainan aplikasi harus memiliki alamat/*address* tersendiri lagi yang disebut *service point address* atau *port address*.
- b. *Segmentation* dan *reassembly*. Sebuah *message* dibagi dalam segmen-segmen yang terkirim. Setiap segmen memiliki *sequence number*. *Sequence number* ini yang berguna bagi *Transport Layer* untuk merakit/*reassembly* segmen-segmen yang terpecah atau terbagi tadi menjadi *message* yang utuh.
- c. *Connection control*. *Transport Layer* dapat berperilaku sebagai *connectionless* atau *connection-oriented*.
- d. *Flow control*. Seperti halnya *Data Link Layer*, *Transport Layer* bertanggung jawab untuk kontrol aliran (*flow control*). Bedanya dengan *flow control* di *Data Link Layer* adalah dilakukan secara *end-to-end*.

- e. *Error control*. Sama fungsi tugasnya dengan *error control* di *Data Link Layer*, namun berorientasi *end-to-end*.

Session Layer

Session Layer bertanggung jawab untuk mengendalikan “*dialog*” antar *node* (*Dialog Control*). Suatu “*dialog*” adalah percakapan formal dimana dua *node* sepakat untuk bertukar data. Untuk melakukan suatu *dialog* maka dibutuhkan tiga fase yaitu:

1. Pembentukan hubungan.

Untuk membentuk kontak, *node* menyepakati aturan-aturan komunikasi, termasuk protokol-protokol yang digunakan dan parameter komunikasinya.

2. Pemindahan data.

Node-node dipakai untuk “*dialog*” pertukaran data.

3. Pemutusan hubungan.

Terjadi ketika *node-node* tidak lagi perlu komunikasi.

Presentation Layer

Presentation Layer bertanggung jawab untuk menyajikan data kepada *Application Layer*. Teknik umum yang digunakan untuk meningkatkan perpindahan data adalah dengan mengubah semua data menjadi format standar sebelum data tersebut dikirimkan. Format standar ini mungkin bukan merupakan format data asli dari setiap komputer. Semua komputer dapat dirancang untuk mengambil data dengan format standar, dan mengubahnya menjadi format data aslinya.

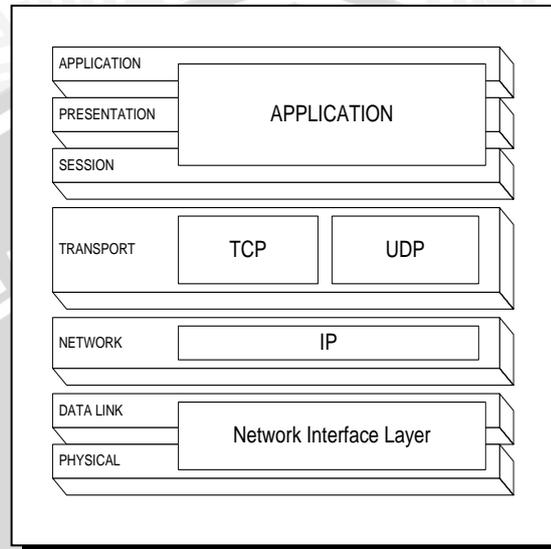
Application Layer

Application Layer menyediakan layanan aplikasi yang digunakan untuk berkomunikasi melalui jaringan. Contoh aplikasi yang diberikan oleh layer ini adalah pengiriman *electronic mail*, *Remote file Access*, dan manajemen jaringan.

2.1.2 *Transmission Control Protocol/Internet Protocol (TCP/IP)*

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di internet.

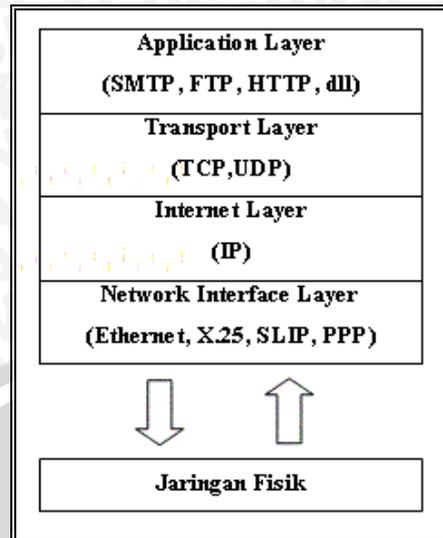
Karena memakai protokol yang sama, maka perbedaan jenis komputer dan sistem operasi tidak menjadi masalah. Protokol TCP/IP hanya dibuat atas empat lapisan saja yaitu *Physical Layer*, *Network Layer*, *Transport Layer* dan *Application Layer*. *Application Layer* pada TCP/IP mencakup tiga lapisan OSI teratas, sebagaimana dapat dilihat pada Gambar 2.2.



Gambar 2.2 Susunan Protokol TCP/IP dan Model OSI
[Sumber : www.prasimax.com]

Masing-masing protokol dalam TCP/IP bertanggung jawab atas bagian-bagian tertentu dari komunikasi data sehingga tugas setiap protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain sepanjang protokol tersebut masih bisa mengirim dan menerima data. Dengan penggunaan prinsip ini, TCP/IP menjadi protokol komunikasi data yang fleksibel.

Dalam TCP/IP, terjadi penyampaian data dari protokol yang berada di satu *layer* ke protokol yang berada di *layer* lain. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data. Jika suatu protokol menerima data dari protokol lain di *layer* di atasnya, ia akan menambahkan informasi tambahan ke data tersebut, kemudian diteruskan ke protokol pada *layer* di bawahnya. Hal sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada *layer* di bawahnya. TCP/IP dimodelkan dengan empat lapisan/*layer* seperti terlihat pada gambar 2.3 berikut :



Gambar 2.3 Layer TCP/IP
[Sumber : Onno W. Purbo, 2003 : 23]

Dari Gambar 2.3 dapat dilihat bahwa TCP/IP terdiri atas empat *layer* yaitu:

1. *Network Interface Layer*
2. *Internet Layer*
3. *Transport Layer*
4. *Application Layer*

Network Interface Layer

Lapisan terbawah adalah *Network Interface Layer*. Lapisan ini disebut juga *Data Link Layer*, bertanggung jawab mengirim dan menerima data ke dan dari media fisik. Media fisiknya dapat berupa kabel, serat optik, atau gelombang radio. Oleh karena itu, lapisan ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer.

Internet Layer

Lapisan berikutnya adalah *Internet Layer* atau *Network Layer*. Protokol yang berada pada lapisan ini bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada lapisan ini terdapat *Internet Protocol (IP)*.

Internet Protocol

Internet Protocol (IP) merupakan inti dari protokol TCP/IP. IP berfungsi untuk menyampaikan paket data ke alamat yang tepat. Seluruh data yang berasal dari protokol pada *layer* di atas IP harus dilewatkan, diolah oleh protokol IP, dan ditransmisikan sebagai paket IP agar sampai ke tujuan. Dalam melakukan pengiriman data, IP memiliki sifat yang dikenal sebagai *unreliable, connectionless, datagram delivery service*.

Unreliable/ketidakhandalan berarti protokol IP tidak menjamin datagram yang dikirim pasti sampai ke tempat tujuan. Protokol IP hanya akan melakukan usaha sebaik-baiknya (*best effort delivery service*) agar paket tersebut sampai ke tempat tujuan. Jika di perjalanan paket tersebut mengalami hal-hal yang tidak diinginkan (seperti salah satu jalur putus, *router* mengalami kongesti/macet, atau *host/network* tujuan sedang *down*), IP hanya mengirim pemberitahuan ke pengirim paket melalui protokol ICMP bahwa terjadi masalah pada pengiriman paket IP ke *host* tujuan.

Connectionless berarti dalam mengirim paket dari tempat asal ke tujuan, IP tidak bertanggung jawab atas proses perjanjian (*handshake*) antara pihak pengirim dan penerima paket IP. *Datagram delivery service* berarti setiap paket data yang dikirim adalah *independent* terhadap paket data yang lain. Akibatnya, jalur yang ditempuh oleh masing-masing paket data IP ke tujuannya bisa jadi berbeda satu dengan lainnya. Karena jalur yang ditempuh berbeda, kedatangan paket pun bisa jadi tidak berurutan.

Transport Layer

Lapisan berikutnya adalah *Transport Layer*. Pada lapisan ini, terdapat protokol yang bertanggung jawab untuk mengadakan komunikasi antara dua *node*. Kedua protokol tersebut ialah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*).

TCP (Transmission Control Protocol)

TCP (*Transmission Control Protocol*), bersifat *connection oriented, reliable, byte stream service*. *Connection oriented* berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan

hubungan (*handshake*) terlebih dahulu. *Reliable* berarti TCP menerapkan proses deteksi kesalahan paket dan melakukan pentransmisian ulang. *Byte stream service* berarti paket dikirimkan dan sampai ke tujuan secara berurutan. Format *header* TCP seperti pada Gambar 2.4 berikut :

| | | | |
|------------------------|------|------------------|--------|
| Source Port | | Destination Port | |
| Sequence Number | | | |
| Acknowledgement Number | | | |
| Hdr | Resv | Control | Window |
| Checksum | | Urgent Pointer | |
| TCP Option | | | |
| Application Data | | | |

Gambar 2.4 Header TCP
[Sumber : Onno W. Purbo, 2002 : 54]

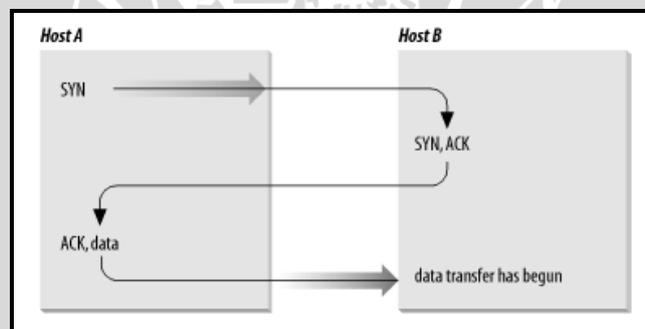
➤ Header TCP terdiri atas beberapa *field* yaitu :

1. *Source Port* (16 bit) dan *Destination Port* (16 bit), mengidentifikasi aplikasi pengirim dan penerima.
2. *Sequence Number* (32 bit), berisi nomor urut *byte stream* dalam data aplikasi yang dikirim.
3. *Acknowledgement Number* (32 bit). Berisi *Sequence Number* berikutnya yang diharapkan penerima setiap kali data sukses dikirim.
4. *Hdr* atau *Header Length* (4 bit), berisi panjang header TCP. Dengan panjang 4 bit, *field* ini harus merepresentasikan panjang *header* TCP dalam satuan 4 byte. Jika 4 bit ini berisi 1 (1111 biner = 15 desimal), maka panjang *header* maksimal adalah $15 \times 4 = 60$ byte.
5. *Resv* atau *Reserve* (6 bit), sebagai cadangan.
6. *Control* (6 bit), adalah bit control yang terdiri atas :
 - a. *Urgent* : 1 = nilai *urgent pointer* adalah valid
 - b. *Ack* : 1 = nilai *acknowledgement* adalah valid
 - c. *PSH* : 1 = *push data*
 - d. *RST* : 1 = koneksi harus di-*reset*
 - e. *SYN* : 1 = Sinkronisasi urutan (*sequence*) selama koneksi

- f. *FIN* : 1 = Putuskan koneksi
7. *Window* (16 bit), berisi panjang *window* (semacam *buffer*) penerima *segment* TCP, merupakan panjang byte maksimal yang bisa diterima setiap saat. Lebar *field* ini adalah 16 bit (2 byte) sehingga nilai maksimalnya adalah 65535.
 8. *Checksum* (16 bit), adalah untuk deteksi *error connection*.
 9. *Urgent Pointer* (16 bit).
 10. *TCP Option*, adalah *header* tambahan dengan jumlah penambahan maksimal sebesar 40 byte.

➤ Pembentukan koneksi

TCP merupakan protokol yang bersifat *connection oriented*. TCP membentuk hubungan secara *end to end* antara dua *host* yang berkomunikasi yang dikenal dengan istilah *three way handshake*. Disebut demikian karena sebelum memulai suatu pengiriman data, TCP akan melakukan pertukaran tiga segmen seperti yang terlihat pada Gambar 2.5 dibawah ini.



Gambar 2.5 Mekanisme *three way handshake*
[Sumber : Craig Hunt, TCP/IP Network Administration, 3rd Edition]

Mekanisme *three way handshake*

- *Host A* memulai hubungan dengan mengirimkan *host B* sebuah segmen *SYN*. Segmen ini memberitahukan kepada *host B* bahwa *Host A* ingin membentuk suatu hubungan, dan juga memberitahukan *sequence number* mana yang akan digunakan oleh *host A* untuk memulai

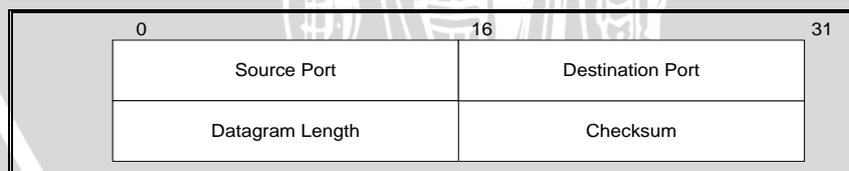
pengiriman segmen datanya (*sequence number* digunakan agar segmen tetap dalam urutannya).

- *Host B* membalas dengan mengirimkan segmen yang berisikan *ACK* dan *SYN*. Segmen ini digunakan untuk memberitahukan bahwa segmen yang dikirim *host A* telah diterima, dan memberitahukan *host A* *sequence number* yang akan dimulai.
- Terakhir, *host A* mengirimkan segmen *ACK* untuk membalas segmen dari *host B* yang diterima dan mengirimkan data pertama

Setelah pertukaran tersebut, *host A* yakin bahwa *host B* terhubung dan siap menerima data. Setelah hubungan terbentuk, data siap dikirimkan. Setelah proses pengiriman data selesai, segment *three way handshake* akan ditukar dengan sebuah segmen yang berisi pesan 'tidak ada lagi data dari pengirim' atau *FIN* segmen untuk memutuskan hubungan.

UDP (*User Datagram Protocol*)

User Datagram Protocol (UDP) merupakan protokol pada *Transport Layer* yang bersifat *connectionless oriented* dan tidak ada *sequencing* (pengurutan kembali) paket yang datang. Gambar 2.6 adalah gambar format *Header* UDP. *Source* dan *destination port* memiliki fungsi yang sama seperti pada *TCP*. *Datagram length* berisi panjang datagram, sedangkan *checksum* berfungsi untuk memeriksa kesalahan data.



Gambar 2.6 *Header* UDP
[Sumber : Stallings, 2000 : 645]

Pada *stack* protokol, antara *layer* satu *layer* dengan *layer* lainnya tidak saling bertanggung jawab. Pada penerapan *IP*, *TCP* dan *UDP* tidak mengalami perubahan karena keduanya berada pada *Transport Layer*, sedangkan *IP* berada pada *Network Layer*.

Application Layer

Layer teratas adalah *Application Layer*. Pada *layer* inilah terletak semua aplikasi yang menggunakan protokol TCP/IP ini. *Layer* ini terdiri atas protokol-protokol tingkat tinggi seperti terminal virtual (TELNET), transfer file (FTP), dan surat elektronik (SMTP). Protokol terminal virtual memungkinkan pengguna pada sebuah mesin untuk *login* ke mesin yang ada di tempat jauh dan bekerja di terminal jarak jauh tersebut. Protokol *transfer file* memungkinkan untuk mengirimkan data dari mesin ke mesin yang lain secara efisien. Surat elektronik pada awalnya hanya merupakan salah satu jenis *transfer file*. Akan tetapi kemudian dibuat protokol khusus untuk keperluan tersebut. Banyak jenis protokol yang telah ditambahkan pada *layer* ini. Misalnya *Domain Name System* (DNS), NNTP, dan banyak lagi protokol-protokol lainnya.

2.2 ROUTER

Router adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai routing. Proses routing terjadi pada lapisan 3 (Lapisan jaringan seperti Internet Protocol) dari stack protokol tujuh-lapis OSI.

Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. *Router* berbeda dengan switch. Switch merupakan penghubung beberapa alat untuk membentuk suatu Local Area Network (LAN).

Analogi Router dan Switch

Sebagai ilustrasi perbedaan fungsi dari *router* dan *switch* merupakan suatu jalanan, dan *router* merupakan penghubung antar jalan. Masing-masing rumah berada pada jalan yang memiliki alamat dalam suatu urutan tertentu. Dengan cara yang sama, *switch* menghubungkan berbagai macam alat, dimana masing-masing alat memiliki alamat IP sendiri pada sebuah LAN.

Router sangat banyak digunakan dalam jaringan berbasis teknologi protokol TCP/IP, dan *router* jenis itu disebut juga dengan *IP Router*. Selain *IP Router*, ada lagi *AppleTalk Router*, dan masih ada beberapa jenis *router* lainnya. Internet merupakan contoh utama dari sebuah jaringan yang memiliki banyak *router IP*. *Router* dapat digunakan untuk menghubungkan banyak jaringan kecil

ke sebuah jaringan yang lebih besar, yang disebut dengan *internetwork*, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya. Router juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda (seperti halnya *router wireless* yang pada umumnya selain ia dapat menghubungkan komputer dengan menggunakan radio, ia juga mendukung penghubungan komputer dengan kabel UTP), atau berbeda arsitektur jaringan, seperti halnya dari Ethernet ke Token Ring.

Router juga dapat digunakan untuk menghubungkan LAN ke sebuah layanan telekomunikasi seperti halnya telekomunikasi leased line atau *Digital Subscriber Line* (DSL). *Router* yang digunakan untuk menghubungkan LAN ke sebuah koneksi leased line seperti T1, atau T3, sering disebut sebagai *access server*. Sementara itu, *router* yang digunakan untuk menghubungkan jaringan lokal ke sebuah koneksi DSL disebut juga dengan *DSL router*. *Router-router* jenis tersebut umumnya memiliki fungsi firewall untuk melakukan penapisan paket berdasarkan alamat sumber dan alamat tujuan paket tersebut, meski beberapa router tidak memilikinya. *Router* yang memiliki fitur penapisan paket disebut juga dengan packet-filtering router. *Router* umumnya memblokir lalu lintas data yang dipancarkan secara broadcast sehingga dapat mencegah adanya *broadcast storm* yang mampu memperlambat kinerja jaringan.

Jenis-jenis router

Secara umum, *router* dibagi menjadi dua buah jenis, yakni:

1. *Static router (router statis)*: adalah sebuah *router* yang memiliki tabel *routing* statis yang diset secara manual oleh para administrator jaringan.
2. *Dynamic router (router dinamis)*: adalah sebuah *router* yang memiliki kemampuan membuat tabel *routing* dinamis, dengan mendengarkan lalu lintas jaringan dan juga dengan saling berhubungan dengan *router* lainnya.

2.3 SMS GATEWAY

SMS Gateway adalah suatu platform yang menyediakan mekanisme untuk UEA menghantar dan menerima SMS dari peralatan mobile (HP, PDA phone, dll) melalui SMS Gateway's shortcode (sbg contoh 9221). Di bawah ini disertakan sedikit ilustrasi mengenai penjelasan di atas.

SMS Gateway membolehkan UEA untuk berkomunikasi dengan Telco SMSC (telkomsel, indosat, dll) atau SMS platform untuk menghantar dan menerima pesan SMS dengan sangat mudah, Karena SMS Gateway akan melakukan semua proses dan koneksi dengan Telco. SMS Gateway juga menyediakan UEA dengan interface yang mudah dan standar.

UEA dapat berupa berbagai aplikasi yang memerlukan penggunaan SMS. Seperti berbagai aplikasi web yang telah banyak menggunakan SMS (free sms, pendaftaran, konfirmasi melalui SMS, aplikasi perkantoran), CMS, acara pengundian di televisi.

UEA melakukan komunikasi dengan SMS Gateway melalui Internet menggunakan standard HTTP GET atau HTTPS (untuk komunikasi yang aman). Telco SMSC akan menghantar pesan (SMS) tersebut kepada perusahaan SMS Gateway (sesuai dengan nomor yang telah disewa) dengan menggunakan protokol yang khusus. Dan berdasarkan keyword yang telah dituliskan pada SMS, maka sistem SMS Gateway akan menghantar SMS tersebut ke URL yang telah ditentukan. UEA dapat menghantar SMS reply kepada pelanggan melalui SMS Gateway tersebut. Dan UEA dapat menentukan besarnya biaya (charging) yang akan dikenakan kepada pelanggan. Biasanya telah ditentukan regulasi biayanya (microcharging mechanism), contoh Rp. 0 (gratis); Rp. 500,- ; Rp. 1000,- ; Rp. 2000,- dst.

Suatu perusahaan SMS Gateway biasanya support untuk pesan yang berupa teks, unicode character, dan juga smart messaging (ringtone, picture message, logo operator,dll).

2.4 TELEPON SELULER

Telepon Seluler (Ponsel) adalah suatu jenis telepon bergerak yang menggunakan teknologi sel sebagai akses komunikasinya sehingga memudahkan seseorang untuk berkomunikasi dimanapun dan dalam kondisi apapun. Sebuah ponsel dapat mengirim dan menerima data suara menggunakan pemancar RF. Dengan adanya ponsel, maka komunikasi lebih mudah dan lebih efisien tetapi sedikit lebih mahal. Seiring perkembangan teknologi di bidang *mobile* sistem, ponsel tidak hanya mengirim data suara tetapi juga data karakter yang biasa disebut SMS (*Short Message Service*). Disamping itu, pesatnya perkembangan

teknologi saat ini menyebabkan ponsel tidak hanya dapat berkomunikasi antara sesama ponsel, tetapi juga dapat berkomunikasi dengan PC (*Personal Computer*). Hal ini menyebabkan kita dapat melakukan pentransferan data antara ponsel dan PC (*Personal Computer*).

2.4.1 Sony Ericsson

Sony Ericsson merupakan salah satu merek ponsel yang dapat berkomunikasi dengan PC (*Personal Computer*) melalui port USB (*Universal Serial Bus*), sehingga suatu pentransferan data dapat terjadi antara PC (*Personal Computer*) dengan ponsel Sony Ericsson tersebut. Melalui pentransferan data ini pengguna ponsel dapat mengirim atau menerima suatu pesan singkat (SMS). Untuk dapat berkomunikasi dengan PC (*Personal Computer*), sony ericsson dilengkapi dengan *internal* modem yang dapat mengenali *AT Command*.

Pin out Telepon Seluler Sony Ericsson W830i

Untuk melakukan koneksi dengan peralatan lain seperti PC (*Personal Computer*) dan mikrokontroler, pada telepon seluler Sony Ericsson tersedia *pinout* yang memiliki fungsi tertentu seperti ditunjukkan pada Tabel 2.1. Dalam Gambar 2.7 ditunjukkan susunan konektor Sony Ericsson W830i.



Gambar 2.7. Susunan *pinout* Sony Ericsson W830i

[Sumber : www.pinouts.ru]

Tabel 2.1. Pin out Konektor Sony Ericsson W830i

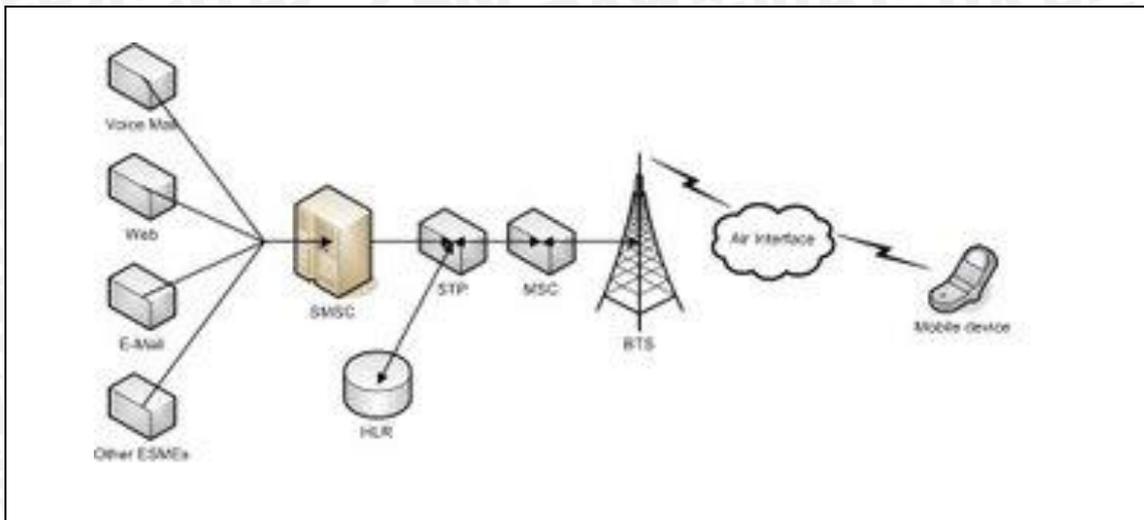
| Pin | Signal | Description |
|-----|--------------|---|
| 1 | USB +5V in | connect to PC USB pin 1 |
| 2 | SP_REF | Audio from mobile/RTS |
| 3 | Mic+/AUXIN_L | Accessory control to mobile |
| 4 | Mic-/AUXIN_R | Accessory control from mobile/handsfree sense |
| 5 | DFMS/SP_L | headphone speaker – left |
| 6 | DTMS/SP_R | headphone speaker – right |
| 7 | VIDEO/STB | video from mobile |
| 8 | VPPFLASH | short to pin 9 for handsfree detection |
| 9 | GND | connect to PC USB pin 4 |
| 10 | USB DATA+ | connect to PC USB pin 3 (Data to mobile (Rx)) |
| 11 | USB DATA- | connect to PC USB pin 2 (Data from mobile (Tx)) |
| 12 | Charge In | should be connected to PC USB pin 1 for charging. |

[Sumber : www.pinouts.ru]

2.5 SHORT MESSAGE SERVICES (SMS)

2.5.1 Umum

Teknologi SMS dimunculkan pertama kali di Eropa sekitar tahun 1991, oleh suatu badan standard Eropa untuk digital wireless yang disebut dengan *Global System For Mobile Communications (GSM)*. SMS menyediakan mekanisme untuk melakukan transmisi pesan pendek dari dan ke telepon genggam. Wireless network memberikan mekanisme yang dibutuhkan untuk menemukan stations tujuan dan mengirimkan short message antara SMSC (*Short Messaging Service Center*) dengan *wireless station*.



Gambar 2.8 Bentuk Jaringan SMS

[Sumber : www.elektronika-elektronika.blogspot.com]

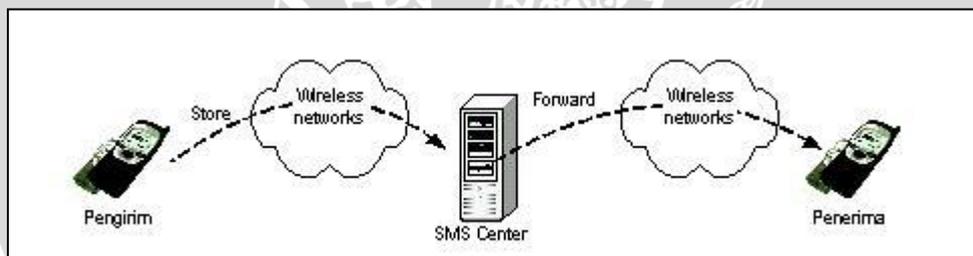
Berikut ini adalah elemen jaringan dan arsitektur dari SMS (Short Message Services) :

- **External Messaging Entities (ESME)** adalah sebuah perangkat yang memungkinkan untuk menerima atau mengirim pesan pendek. Sebuah *short message entity* (SME) harus ditaruh di dalam jaringan, *mobile device* atau *service center* yang lain.
- **SMS Centre (SMSC)** adalah kombinasi dari *hardware* dan *software* yang bertanggung jawab untuk menyampaikan dan menyimpan pesan antara SME dan *mobile device*. SMSC harus mempunyai reliabilitas yang tinggi sehingga dapat mengakomodasi permintaan SMS yang semakin berkembang.
- **Signal Transfer Point (STP)** adalah elemen jaringan yang tersedia pada deployment IN (*Intelligent Network*) untuk menghubungkan *link* dengan elemen jaringan yang *multiple*.
- **Home Location Register (HLR)** adalah database yang digunakan untuk penyimpanan permanen dan manajemen *subscription* dan *service profile*. jika station tujuan tidak ada maka HLR akan menginformasikan pada SMSC bahwa *station* diperlukan untuk diakses oleh jaringan *mobile*.
- **Visitor Location Register (VLR)** adalah database yang berisi informasi sementara tentang *subscriber* yang masuk.

- **MSC** membentuk fungsi *switch* dari sistem dan melakukan kontrol panggilan dari dan ke telepon serta sistem data. MSC akan menyampaikan data pesan ke mobile *subscriber* yang dituju melalui *base station*.
- **Air Interface** didefinisikan pada setiap satu teknologi *wireless* yang berbeda (GSM,TDMA,CDMA). *Air interface* menspesifikasikan bagaimana suara atau signal data ditransfer dari MSC ke *handset*.
- **Base Station System** terdiri dari *base station controllers* (BSCs) dan *Base transceiver stations* (BTSs) yang berfungsi untuk mengatur transmisi elektromagnetik antar MSC dan *mobile devices*.
- **Mobile Device** terminal *wireless* yang mampu untuk menerima dan mengirimkan pesan SMS melalui jaringan *wireless*.

2.5.2 Konsep Dasar Pengiriman SMS

Ketika mengirim pesan SMS melalui handphone (*mobile originated*), pesan tidak langsung dikirimkan ke handphone tujuan (*mobile terminated*), akan tetapi pesan dikirim terlebih dahulu ke SMS Center (SMSC) yang kemudian diteruskan ke handphone tujuan.



Gambar 2.9 Skema Konsep Dasar Pengiriman SMS
[Sumber : RAY-01:11]

Pada SMSC, dapat diketahui status dari pesan SMS yang telah dikirim, apakah telah sampai atau gagal diterima oleh handphone tujuan. Apabila handphone tujuan dalam keadaan aktif dan dapat menerima pesan SMS yang dikirim, handphone tujuan akan mengirimkan kembali pesan konfirmasi ke SMSC yang menyatakan bahwa pesan telah diterima. Kemudian SMSC mengirimkannya kembali status tersebut kepada pengirim. Jika handphone tujuan dalam keadaan mati, pesan yang dikirimkan akan disimpan pada SMSC sampai *validity-period* terpenuhi.

2.5.3 Format Data SMS (*Short Message Service*)

SMS (*Short Message Service*) merupakan sebuah mekanisme pengiriman pesan singkat melalui jaringan bergerak (*mobile network*). Panjang maksimum dari sebuah pesan singkat adalah 160 karakter, fasilitas ini disediakan oleh jaringan telepon seluler. Sebenarnya panjang pesan maksimum yang dapat dikirimkan melalui SMS adalah 140 karakter. Teknik ini bertumpu pada keadaan bahwa kode karakter ASCII alfanumerik yang mempunyai lebar data 7 bit (bit ke-7 selalu bernilai 0 sehingga bisa diabaikan). Teknik kompresi *septet to oktet* dilakukan dengan menyisipkan bit-bit LSB karakter selanjutnya ke dalam bit-bit MSB dari data sebelumnya secara berkesinambungan.

Ada dua cara pengiriman dan penerimaan SMS, yaitu dengan menggunakan mode teks dan mode PDU (*Protocol Data Unit*). Mode teks tidak terdapat pada beberapa telepon seluler, maka dalam skripsi ini digunakan pesan SMS dengan mode PDU dan tidak membahas mode yang lain. PDU berisi bilangan-bilangan heksadesimal yang terdiri atas: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

2.5.4 Prinsip Kerja SMS (*Short Message Service*)

Dibalik tampilan menu-menu SMS dalam telepon seluler sebenarnya adalah *AT Command* yang bertugas mengirim, menerima dan menghapus data dari dan ke SMS *centre*. *AT Command* tiap-tiap perangkat SMS bisa berbeda-beda tergantung dari jenis telepon seluler, akan tetapi pada dasarnya sama berdasarkan fungsinya. Data-data yang mengalir ke dan dari *SMS-centre* berbentuk PDU (*Protocol Data Unit*). PDU ini berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa input output. PDU tersebut terdiri dari beberapa *header*. *Header* untuk mengirim SMS ke *SMS-centre* berbeda dengan SMS yang diterima dari *SMS-centre*.

PDU untuk kirim SMS ke SMS-Centre

Fomat PDU untuk mengirim SMS terdiri atas delapan *header*, yaitu sebagai berikut :

1. Nomor SMS-Centre

Header pertama ini terbagi atas tiga *subheader*, yaitu :

- a. Jumlah pasangan heksadesimal SMS-Centre dalam bilangan heksa.
- b. Kode *National/International*.
 - Untuk *National*, kode *subheader*-nya yaitu 81
 - Untuk *International*, kode *subheader*-nya yaitu 91
- c. Nomor SMS-Centre-nya sendiri, dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F di depannya. Contoh: untuk nomor SMS-Centre Telkomsel dapat ditulis dengan 2 cara yaitu :
 - Cara 1, 0818445009 diubah menjadi:
 - 1) 06 → ada 6 pasang
 - 2) 81 → 1 pasang
 - 3) 80-81-44-05-90 → 5 pasang
 Total ada 6 pasang. Di gabung menjadi : 06818081440590
 - Cara 2, 62818445009 diubah menjadi:
 - 1) 07 → ada 7 pasang
 - 2) 91 → 1 pasang
 - 3) 26-18-48-54-00-F9 → 6 pasang
 Total ada 7 pasang. Digabung menjadi : 07912618485400F9

Berikut ini beberapa nomor SMS-Centre operator seluler di Indonesia :

Tabel 2.2 Beberapa nomor SMS-Centre *National Code*

| No | Operator Seluler | SMS-Centre No | Kode PDU |
|----|------------------|---------------|----------------|
| 1. | Telkomsel | 0811000000 | 068180110000F0 |
| 2. | Satelindo | 0816125 | 0581806121F5 |
| 3. | Excelcom | 0818445009 | 06818081440590 |
| 4. | Indosat-M3 | 0855000000 | 06818055000000 |

[Sumber : Bustam, 2002 : 10]

Tabel 2.3 Beberapa nomor SMS-Centre International Code

| No | Operator Seluler | SMS-Centre No | Kode PDU |
|----|------------------|---------------|------------------|
| 1. | Telkomsel | 62811000000 | 06912618010000 |
| 2. | Satelindo | 62816124 | 059126181642 |
| 3. | Excelcom | 62818445009 | 07912618485400F9 |
| 4. | Indosat-M3 | 62855000000 | 07912658050000F0 |

[Sumber : Bustam, 2002 : 11]

2. Tipe SMS

Untuk tipe SMS kirim = 1, jadi bilangan heksanya adalah 01

3. Nomor Referensi

Nomor referensi ini dibiarkan dulu 0, jadi bilangan heksanya adalah 00.

Nomor referensi ini akan diberikan secara otomatis oleh ponsel tersebut.

4. Nomor Ponsel Penerima

Header ini juga terbagi menjadi 3 bagian :

- a. Jumlah bilangan desimal nomor ponsel yang dituju dalam bilangan heksa
- b. *National/International Code*
 - Untuk *National*, kode *subheader*-nya yaitu 81
 - Untuk *International*, kode *subheader*-nya yaitu 91
- c. Nomor ponsel yang dituju, dalam pasangan heksa dibalik. Jika tertinggal satu angka heksa atau tidak memiliki pasangan, angka tersebut dipasangkan dengan huruf F di depannya.

Contoh untuk nomor ponsel yang dituju 628129573337 dapat ditulis dengan dua cara, yaitu sebagai berikut :

- Cara 1, 08129573337 diubah menjadi :
 - a. 0B → ada 11 angka
 - b. 81
 - c. 80-21-59-37-33-F7

Sehingga digabung menjadi : 0B818021593733F7

- Cara 2, 628129573337 diubah menjadi :
 - a. 0C → ada 12 angka
 - b. 91
 - c. 26-18-92-75-33-73

Sehingga digabungkan menjadi : 0C91261892753373

5. Bentuk SMS, antara lain :

0 → 00 → dikirim sebagai SMS.

1 → 01 → dikirim sebagai *telex*.

2 → 02 → dikirim sebagai *fax*.

Dalam hal ini, untuk mengirim dalam bentuk SMS tentu saja harus memakai kode heksa 00.

6. Skema *Encoding Data I/O*

Ada dua skema, yaitu :

a. Skema 7 bit, ditandai dengan angka 0 → 00

b. Skema 8 bit, ditandai dengan angka lebih besar dari 0 kemudian diubah ke heksa.

Kebanyakan ponsel/SMS *Gateway* yang ada di pasaran sekarang menggunakan skema 7 bit sehingga kita menggunakan kode 00

7. Jangka waktu/waktu validitas

Jika jangka waktu validitas diabaikan, itu berarti tidak membatasi waktu berlakunya SMS. Sedangkan jika diisi dengan suatu bilangan integer yang kemudian diubah ke pasangan heksa tertentu, bilangan yang diberikan tersebut akan mewakili jumlah waktu validitas SMS tersebut. Rumus untuk menghitung jangka waktu validitas SMS seperti pada tabel 2.4 sebagai berikut :

Tabel 2.4 Rumus untuk Menghitung Jangka Waktu Validitas SMS

| Integer (INT) | Jangka Waktu Validitas SMS |
|---------------|---|
| 0 – 143 | $(INT + 1) \times 5$ menit (berarti : 5 menit s/d 12 jam) |
| 144 – 167 | 12 jam + $((INT - 143) \times 30)$ menit |
| 168 – 196 | $(INT - 166) \times 1$ hari |
| 197 – 255 | $(INT - 192) \times 1$ minggu |

[Sumber : Bustam, 2002 : 13]

8. Isi SMS

Header ini terdiri atas dua *subheader*, yaitu :

- a. Panjang isi (jumlah huruf dari isi)

Misalnya : untuk kata “hello”, ada 5 huruf maka penulisannya 05.

- b. Isi berupa pasangan bilangan heksa

Untuk ponsel/SMS *gateway* berskema *encoding* 7 bit, jika mengetik suatu huruf dari keypadnya, berarti telah membuat 7 angka I/O berturutan. Skema 7 bit tersebut diperlihatkan pada Tabel 2.5.

Ada 2 langkah yang harus dilakukan untuk mengkonversikan isi SMS, yaitu:

Langkah pertama : mengubah menjadi kode 7 bit.

Langkah kedua : mengubah kode 7 bit menjadi 8 bit, yang diwakili oleh pasangan heksa.

Contoh untuk kata “hello”

Langkah pertama :

| Bit | 7 | 1 |
|-----|-----|------|
| h | 110 | 1000 |
| e | 110 | 0101 |
| l | 110 | 1100 |
| l | 110 | 1100 |
| o | 110 | 1111 |

Langkah Kedua:

| | | | |
|---|---------------|------|------|
| | E | 8 | |
| h | <u>1</u> 110 | 1000 | |
| | 3 | 2 | |
| e | <u>00</u> 11 | 0010 | 1 |
| | 9 | B | |
| l | <u>100</u> 1 | 1011 | 00 |
| | F | D | |
| l | <u>1111</u> | 1101 | 100 |
| | 0 | 6 | |
| o | <u>0000</u> 0 | 110 | 1111 |

Oleh karena total 7 bit x 5 huruf = 35 bit, sedangkan yang kita perlukan adalah 8 bit x 5 huruf = 40 bit, maka diperlukan 5 bit *dummy* (sisanya) yang diisi dengan bilangan 0. Setiap 8 bit mewakili suatu pasangan heksa. Tiap 4 bit mewakili suatu angka heksa. Dengan demikian kata “hello” hasil konversinya menjadi E8329BFD06.

Tabel 2.5 Skema 7 Bit SMS pada Telepon Seluler

| | | | | b7 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|-----------|-----------|-----------|-----------|-----------|----|---|----|---|---|---|---|---|
| | | | | b6 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | | | | b5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| b4 | b3 | b2 | B1 | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 0 | 0 | 0 | @ | Δ | SP | 0 | - | P | ¨ | p |
| 0 | 0 | 0 | 1 | 1 | | | ! | 1 | A | Q | a | q |
| 0 | 0 | 1 | 0 | 2 | \$ | Φ | " | 2 | B | R | b | r |
| 0 | 0 | 1 | 1 | 3 | | Γ | # | 3 | C | S | c | s |
| 0 | 1 | 0 | 0 | 4 | | Λ | | 4 | D | T | d | t |
| 0 | 1 | 0 | 1 | 5 | | Ω | % | 5 | E | U | e | u |
| 0 | 1 | 1 | 0 | 6 | | Π | & | 6 | F | V | f | v |
| 0 | 1 | 1 | 1 | 7 | | Ψ | . | 7 | G | W | g | w |
| 1 | 0 | 0 | 0 | 8 | | Σ | (| 8 | H | X | h | x |
| 1 | 0 | 0 | 1 | 9 | | Θ |) | 9 | I | Y | i | y |
| 1 | 0 | 1 | 0 | 10 | LF | Ξ | * | : | J | Z | j | z |
| 1 | 0 | 1 | 1 | 11 | | | + | ; | K | Ä | k | ä |
| 1 | 1 | 0 | 0 | 12 | | | , | < | L | Ö | l | ö |
| 1 | 1 | 0 | 1 | 13 | CR | | - | = | M | | m | |
| 1 | 1 | 1 | 0 | 14 | | β | . | > | N | Ü | n | ü |
| 1 | 1 | 1 | 1 | 15 | | | / | ? | O | | o | |

[Sumber : Bustam, 2002 : 15]

PDU untuk SMS Terima dari SMS-Centre

Terdapat 8 *header* untuk menerima SMS yang pada prinsipnya hampir sama dengan 8 *header* untuk mengirim SMS. *Header* tersebut meliputi:

1. Nomor SMS-Centre
2. Tipe SMS untuk SMS-terima = 4 menjadi 04
3. Nomor ponsel pengirim
4. Bentuk SMS
5. Skema *Encoding*
6. Tanggal dan waktu SMS di SMS-Centre

Diwakili oleh 12 bilangan heksa (6 pasangan) yang berarti : yy/mm/dd hh:mm:ss. Misalnya : 207022512380 yang berarti 02/07/22 15:32:08 atau 22 Juli 2002 15:32:08 WIB.

7. Batas waktu validitas, jika tidak dibatasi dilambangkan dengan 00
8. Isi SMS.

Contoh untuk kode PDU sebagai berikut 07912658050000F0,04,0C91265816107398,00,00,207022512380,00,05E8329BFD06. Dapat kita artikan sebagai berikut :

1. SMS tersebut dikirim lewat SMS-Centre : 62855000000.
2. SMS tersebut merupakan SMS terima.
3. SMS tersebut dikirim dari ponsel nomor 628561013789.
4. SMS tersebut diterima dalam bentuk SMS.
5. SMS tersebut memiliki skema *encoding* 7 bit.
6. SMS tersebut sampai di SMS-Centre pada tanggal 22-07-02, pukul : 15:32:08 WIB.
7. SMS tersebut tidak memiliki batas waktu valid.
8. SMS tersebut isinya adalah "hello".

2.6 AT Command

Komunikasi data antara telepon seluler dengan periperal lain seperti mikrokontroler dilakukan secara serial menggunakan perintah-perintah AT (*AT Command*). Dengan mengirimkan perintah-perintah AT yang spesifik dapat memerintahkan telepon seluler untuk melakukan apa yang kita inginkan.

Tabel 2.6 berikut merupakan beberapa perintah-perintah AT yang digunakan untuk mengoperasikan SMS.

Tabel 2.6 AT Command pada SMS

| Perintah | Fungsi |
|----------|---------------------------------------|
| AT+CMGC | Mengirim sebuah perintah SMS |
| AT+CMGD | Menghapus sebuah SMS dalam SMS memori |
| AT+CMGF | SMS Format |
| AT+CMGL | Daftar SMS |
| AT+CMGR | Membaca dalam sebuah SMS |
| AT+CMGS | Mengirim sebuah SMS |
| AT+CSCA | Alamat dari pusat SMS servis |

[Sumber: www.SonyEricsson.com]

2.7 Performansi SMS (*Short Message Service*)

Salah satu faktor penting dalam pengukuran performansi jaringan adalah *delay*.

2.7.1 *Delay end to end*

Delay adalah waktu yang dibutuhkan untuk mengirimkan data dari sumber sampai tujuan. Pada implementasi *Short Message Service* pada jaringan GSM untuk menganalisa performansi *delay* jaringan digunakan *delay end to end*

Delay end to end adalah *delay* yang terjadi dalam perjalanan paket data dari sumber ke tujuannya setiap *node*. Untuk memperhitungkan *delay end-to-end*, mengacu pada gambar 2.10, dimana pada perhitungannya *delay end to end* yang dianalisa meliputi *delay* transmisi, *delay* propagasi, dan *delay* antrian. Persamaan *delay end to end* adalah sebagai berikut :

$$D_{\text{end-to-end}} = DT(\text{total}) + DP(\text{total}) + T_{\text{queue}} \dots \dots \dots (2-1)$$

dimana:

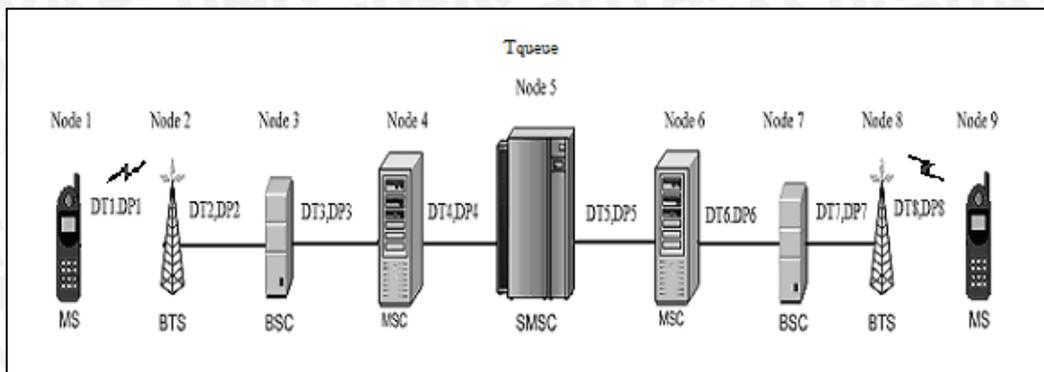
$D_{\text{end-to-end}}$ = *Delay* total dari MS pengirim ke MS penerima. (detik)

$DT(\text{total})$ = *Delay* transmisi total dari MS pengirim ke MS penerima(detik).

$DP(\text{total})$ = *Delay* propagasi total dari MS pengirim ke MS penerima (detik).



T_{queue} = Delay antrian yang terjadi di SMSC (detik).



Gambar 2.10 Delay end-to-end SMS pada jaringan GSM [Sumber: Analisis]

2.7.2 Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk mentransmisikan sebuah paket data. Delay transmisi untuk tiap-tiap node dapat dihitung dengan menggunakan persamaan:

$$DT = \frac{L}{C} \dots\dots\dots (2-2)$$

Besarnya delay transmisi total untuk layanan short message pada jaringan GSM menggunakan persamaan :

$$DT(\text{total}) = DT1 + DT2 + DT3 + DT4 + DT5 + DT6 + DT7 + DT8 \dots\dots\dots (2-3)$$

dimana :

DT(total) = Delay transmisi total dari MS pengirim ke MS penerima (detik).

DT1 = Delay transmisi dari MS pengirim ke BTS (detik).

DT2 = Delay transmisi dari BTS ke BSC (detik).

DT3 = Delay transmisi dari BSC ke MSC (detik).

DT4 = Delay transmisi dari MSC ke SMSC (detik).

DT5 = Delay transmisi dari SMSC ke MSC (detik).

DT6 = Delay transmisi dari MSC ke BSC (detik).

DT7 = Delay transmisi dari BSC ke BTS (detik).

DT8 = Delay transmisi dari BTS ke MS penerima (detik).

L = panjang paket data (bit)

C = kecepatan transmisi (bps)



2.7.3 Delay Propagasi

Delay propagasi adalah waktu perambatan atau penjalaran yang dibutuhkan oleh data dari suatu *node* ke *node* yang lain melalui media transmisi.

Delay propagasi dapat dihitung menggunakan persamaan :

$$DP = \frac{d}{c} \dots\dots\dots(2-4)$$

Dan besarnya *delay* propagasi total dapat dihitung menggunakan persamaan :

$$DP(\text{total}) = DP1 + DP2 + DP3 + DP4 + DP5 + DP6 + DP7 + DP8 \dots\dots\dots(2-5)$$

dimana :

DP(total) = *Delay* propagasi total dari MS pengirim ke MS penerima (detik)

DP1 = *Delay* propagasi dari MS pengirim ke BTS (detik).

DP2 = *Delay* propagasi dari BTS ke BSC (detik).

DP3 = *Delay* propagasi dari BSC ke MSC (detik).

DP4 = *Delay* propagasi dari MSC ke SMSC (detik).

DP5 = *Delay* propagasi dari SMSC ke MSC (detik).

DP6 = *Delay* propagasi dari MSC ke BSC (detik).

DP7 = *Delay* propagasi dari BSC ke BTS (detik).

DP8 = *Delay* propagasi dari BTS ke MS penerima (detik).

d = Jarak tiap-tiap *node*

c = Kecepatan propagasi ($3 \cdot 10^8$ m/s)

2.7.4 Delay Antrian

Delay antrian adalah waktu dimana paket data tersebut berada dalam antrian untuk ditransmisikan. Selama waktu ini paket data menunggu sampai selesainya paket lain ditransmisikan. Parameter yang diperlukan dalam *delay* antrian adalah panjang paket data (L), kecepatan transmisi (C), kecepatan pesan yang datang (λ), dan faktor kegunaan (ρ).

Delay antrian total adalah penjumlahan dari waktu tunggu dalam antrian (t_{queue}) dan waktu pelayanan (t_s) dalam router, sehingga *delay* antrian dapat dirumuskan seperti pada persamaan :

$$T_{\text{queue}} = t_{\text{queue}} + t_s \dots\dots\dots(2-6)$$



Waktu pelayanan:

$$t_s = 1/\mu = \frac{L}{C} \dots\dots\dots(2-7)$$

Besarnya kecepatan pesan yang datang:

$$\lambda = \rho \mu \dots\dots\dots(2-8)$$

Rata-rata waktu tunggu dalam antrian (t_{queue})

$$t_{queue} = \frac{\lambda}{\mu^2 (1-\rho)} \dots\dots\dots(2-9)$$

dimana :

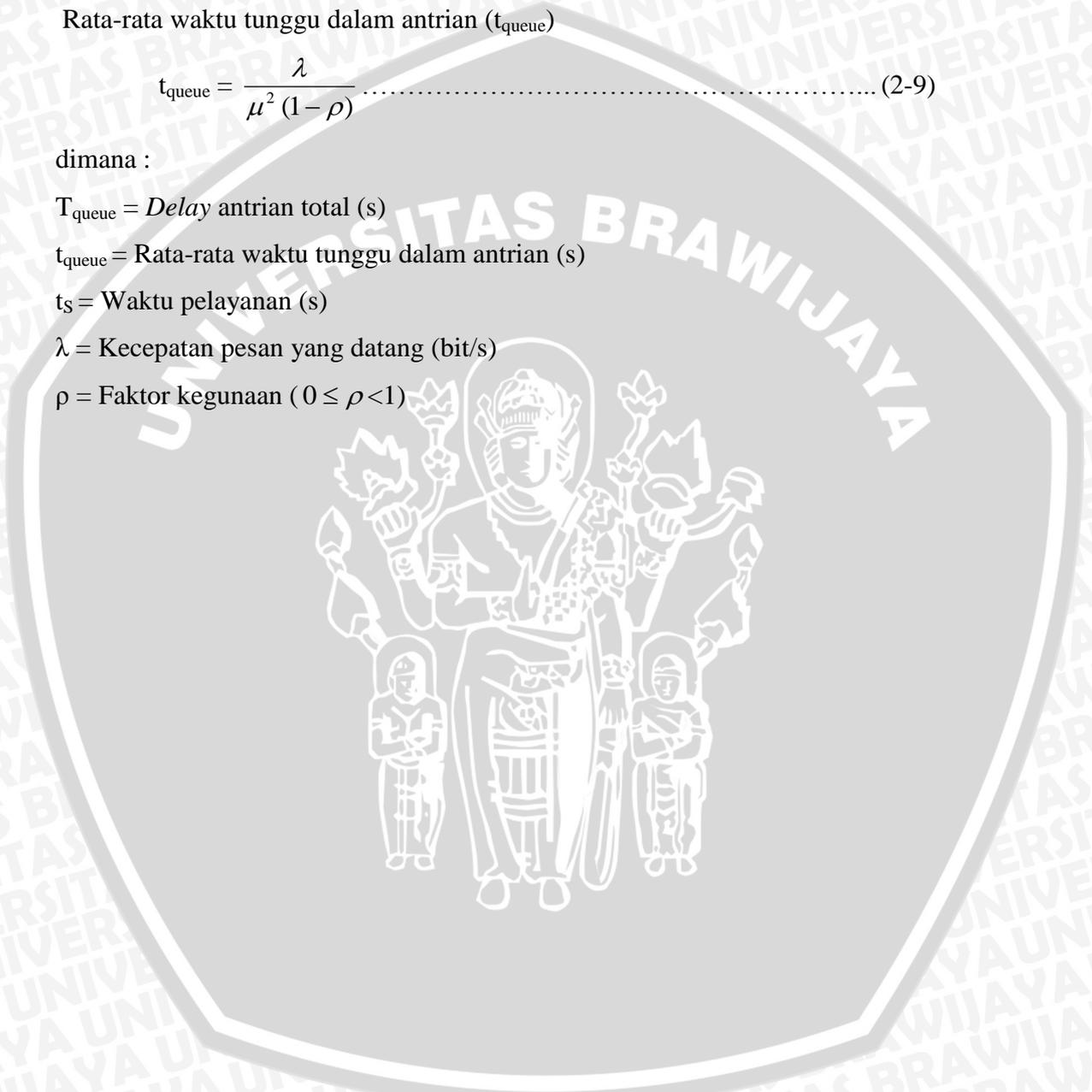
T_{queue} = Delay antrian total (s)

t_{queue} = Rata-rata waktu tunggu dalam antrian (s)

t_s = Waktu pelayanan (s)

λ = Kecepatan pesan yang datang (bit/s)

ρ = Faktor kegunaan ($0 \leq \rho < 1$)



BAB III

METODOLOGI PENELITIAN

Kajian dalam skripsi ini bersifat perancangan yaitu perancangan sistem monitoring koneksi *router* di Fakultas Teknik Universitas Brawijaya menggunakan *SMS Gateway*. Tahapan yang dilakukan untuk solusi permasalahan sesuai dengan topik penelitian adalah menentukan jenis data dan cara mendapatkannya, variabel dan cara analisis yang dilakukan serta rangka solusi terhadap seluruh tahapan dalam bentuk diagram alir.

3.1 Jenis Data dan Cara Mendapatkan Data

3.1.1 Data Primer

Data primer adalah data yang diambil berdasarkan studi lapangan berupa pengamatan di lapangan. Data ini diambil dari operator seluler GSM dan TPTI Fakultas Teknik Universitas Brawijaya. Data primer yang diperoleh adalah :

1. Pengambilan data pada operator seluler GSM berupa data kecepatan transmisi jarak saluran trasmisi, dan *delay end-to-end*.
2. Pengambilan data dari TPTI Fakultas Teknik Universitas Brawijaya berupa gambar topologi jaringan Fakultas Teknik Universitas Brawijaya.

3.1.2 Data Sekunder

Data sekunder berupa studi literatur yang diperoleh dari berbagai buku, jurnal-jurnal dan dari internet yang berhubungan dengan perancangan sistem monitoring koneksi *router*. Beberapa data sekunder yang diperoleh adalah :

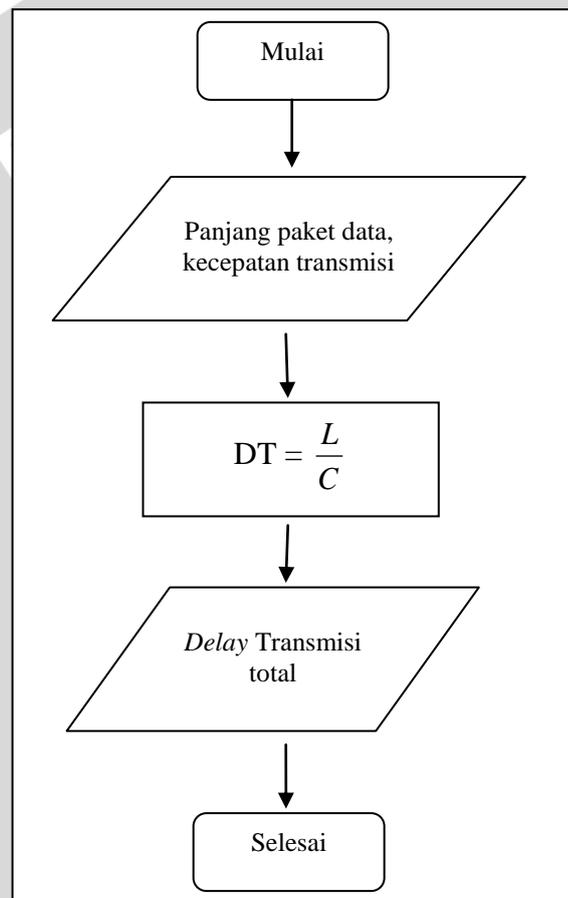
1. Sistem kerja SMS, *AT-Command*, format data PDU (*Protocol Data Unit*).
2. Kajian antarmuka *handphone* dan komputer, serta komunikasi data USB (*Universal Serial Bus*) yang akan digunakan dalam perancangan.
3. Kajian pustaka mengenai Kannel yang digunakan sebagai *SMS Gateway*.
4. Rumus perhitungan *delay end-to-end* beserta parameter-parameter yang digunakan dalam perhitungan.

3.2 Variabel dan Cara Analisis Data yang Digunakan

Pada metode ini dilakukan proses analisis data dari variabel – variabel yang digunakan dalam penghitungan. Variabel tersebut meliputi:

3.2.1 Variabel *Delay Transmisi*

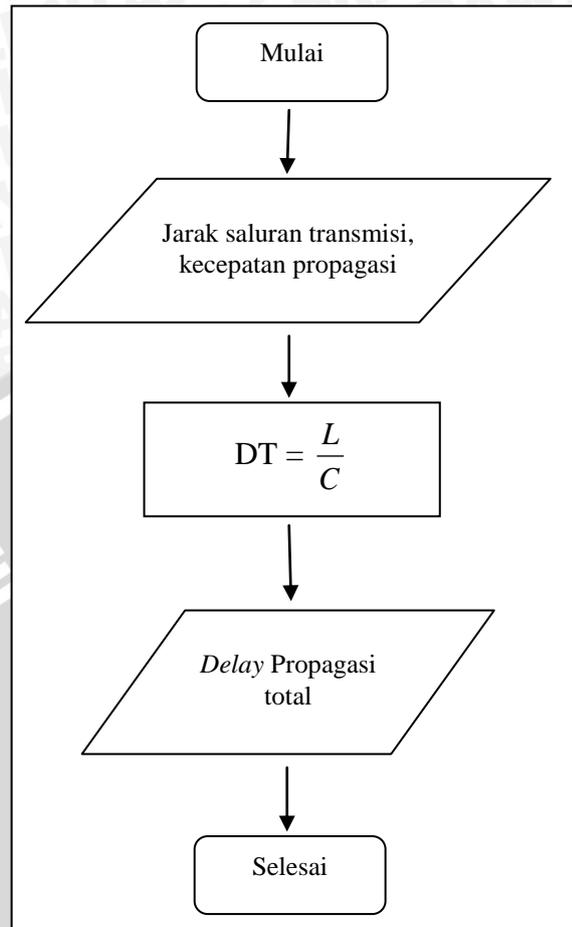
Delay transmisi adalah waktu yang dibutuhkan untuk mentransmisikan sebuah paket data. *Delay transmisi* untuk tiap-tiap *node* dapat dihitung dengan menggunakan persamaan (2-2). Besarnya *delay transmisi* total untuk layanan *short message* pada jaringan GSM menggunakan persamaan (2-3).



Gambar 3.1 Penghitungan *Delay Transmisi*
[Sumber : Perancangan]

3.2.2 Variabel *Delay Propagasi*

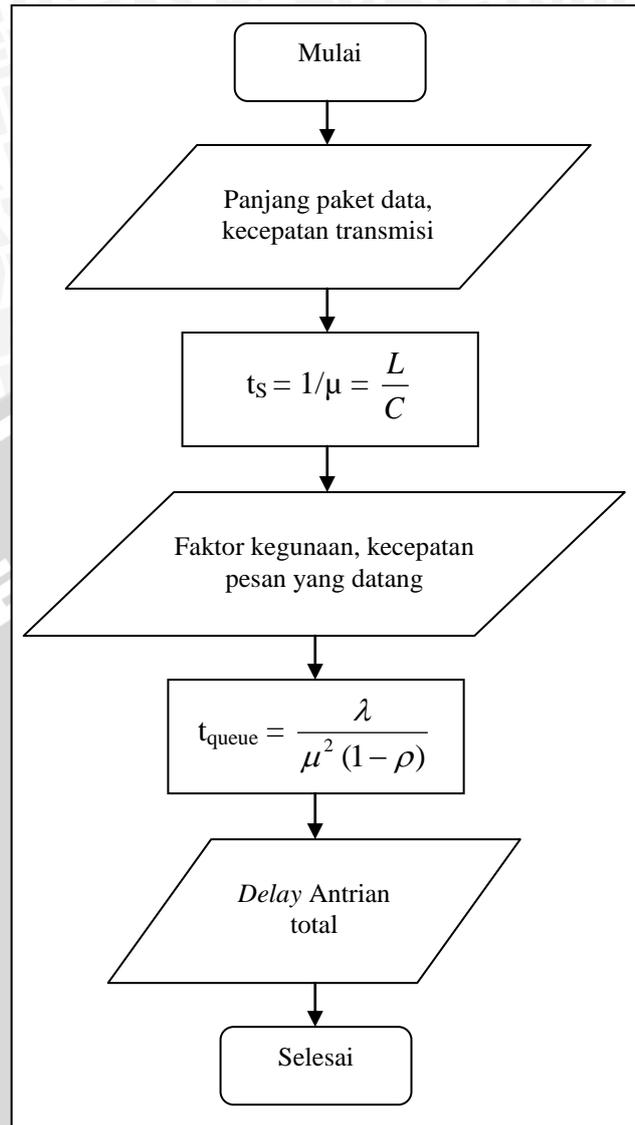
Delay propagasi adalah waktu perambatan atau penjalaran yang dibutuhkan oleh data dari suatu *node* ke *node* yang lain melalui media transmisi. *Delay propagasi* dapat dihitung menggunakan persamaan (2-4) dan besarnya *delay propagasi* total dapat dihitung menggunakan persamaan (2-5).



Gambar 3.2 Penghitungan *Delay Propagasi*
[Sumber : Perancangan]

3.2.3 Variabel *Delay Antrian*

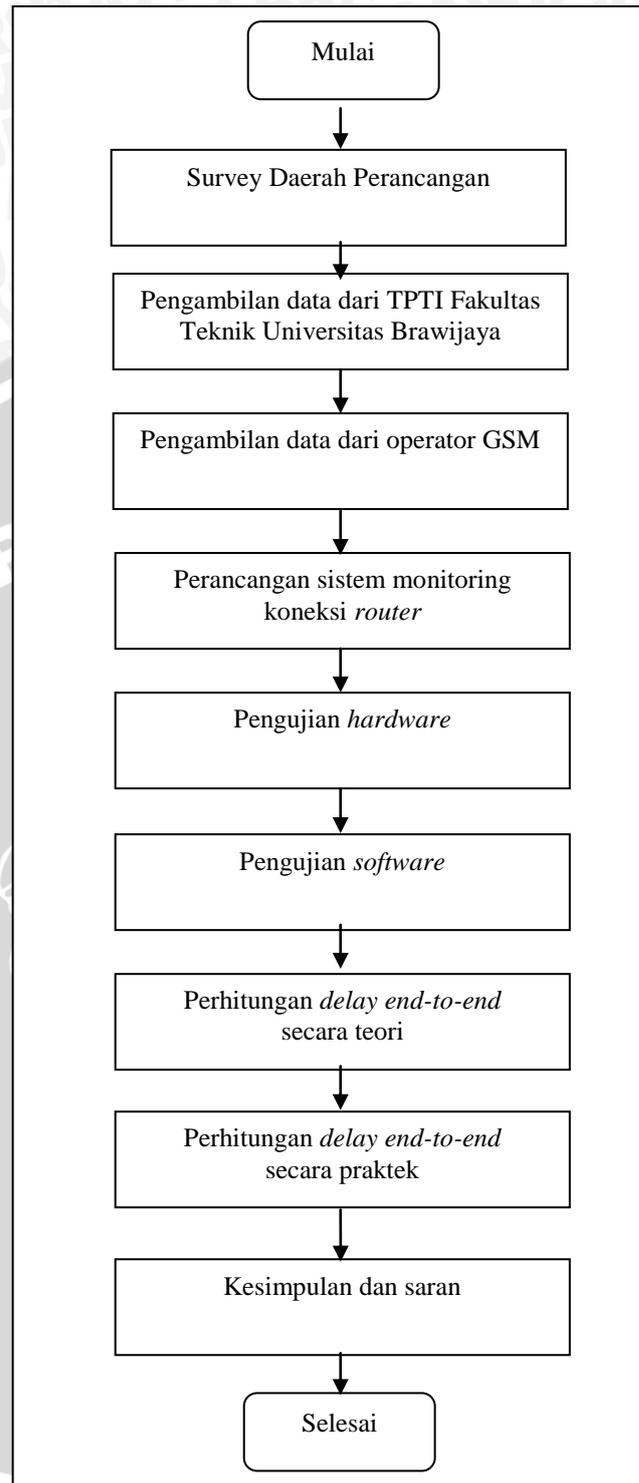
Delay antrian adalah waktu dimana paket data tersebut berada dalam antrian untuk ditransmisikan. Selama waktu ini paket data menunggu sampai selesainya paket lain ditransmisikan. *Delay antrian total* adalah penjumlahan dari waktu tunggu dalam antrian (t_{queue}) dan waktu pelayanan (t_s) dalam router, sehingga *delay antrian* dapat dirumuskan seperti pada persamaan (2-6).



Gambar 3.3 Penghitungan *Delay* Antrian
[Sumber : Perancangan]

3.3 Rangka Solusi Permasalahan

Rangka solusi permasalahan yang dimaksudkan adalah tahapan solusi permasalahan sesuai dengan rumusan masalah, aspek kajian dan tujuan yang telah ditetapkan. Rangka solusi permasalahan diuraikan dalam bentuk flowchart dan penjelasan.

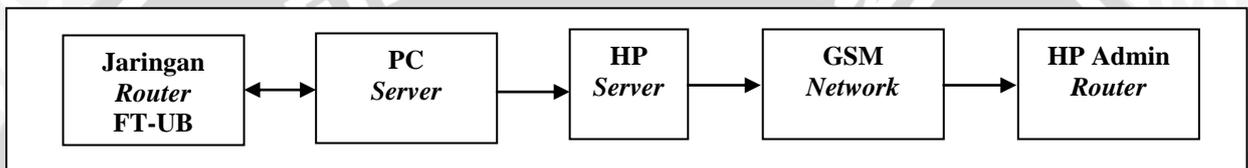


Gambar 3.4 Diagram Alir Solusi Permasalahan
[Sumber : Perancangan]

1. Survey Daerah Perancangan

Survey dilakukan untuk mengetahui daerah yang nantinya akan dirancang sistem monitoring koneksi *router*.

2. Pengambilan Data dari TPTI Fakultas Teknik Universitas Brawijaya
Data yang di ambil dari TPTI Fakultas Teknik Universitas Brawijaya adalah berupa gambar topologi jaringan Fakultas Teknik Universitas Brawijaya..
3. Pengambilan data dari operator seluler GSM
Pengambilan data pada operator seluler GSM berupa data kecepatan transmisi jarak saluran trasmisi, dan *delay end-to-end*.
4. Perancangan Sistem
Perencanaan sistem meliputi perancangan dan konfigurasi program yang digunakan untuk SMS *Gateway*. Diagram blok dari sistem yang akan digunakan sebagai berikut :



Gambar 3.5 Diagram blok sistem
[Sumber : Perancangan]

- a. *Personal Computer* (PC) digunakan sebagai komputer *server*, yang berfungsi sebagai pelayanan jaringan, mengatur lalu lintas data dalam sebuah jaringan, dan menyediakan *resource* yang dapat dipakai oleh komputer lain yang terhubung dalam jaringannya. Pada *Personal Computer* tersebut dihubungkan dengan *handphone* Sony Ericsson W830i, yang berfungsi untuk mengirimkan pesan bahwa keadaan koneksi *router* dalam keadaan terhubung ke *intranet* atau tidak. Dalam hal ini menggunakan kabel *Universal Serial Bus* (USB) DCU-60 sebagai penghubung komputer dengan *handphone*.

Spesifikasi *Server*:

- a. Intel Pentium 4 cpu 2,4 GHz
- b. Memory DDR 256 MB
- c. Lan Card : VIA Rhine II 100Mbps full-duplex
- d. Port USB (*Universal Serial Bus*)
- b. GSM *Network* berfungsi sebagai media pengiriman SMS dari *handphone* (HP) yang terhubung dengan *personal computer* (PC) ke *handphone* admin komputer *router*.

- c. *Handphone* (HP) admin komputer *router* berfungsi sebagai penerima pesan yang dikirim oleh *handphone* yang terhubung dengan *personal computer* (PC).
- d. *Handphone* (HP) *server* berfungsi sebagai pengirim pesan yang berasal dari *personal computer* (PC) *server* SMS Gateway. *Handphone* (HP) *server* menggunakan *handphone* Sony Ericsson W830i.
- e. Dalam perancangan sistem tersebut digunakan aplikasi SMS Gateway Kannel.

5. Pengujian *hardware*

Pengujian *hardware* (perangkat keras) adalah menguji hubungan komunikasi antara *handset* GSM dengan PC, sehingga sistem monitoring koneksi *router* dapat berjalan, antara lain *baudrate* dan waktu transmisi.

6. Pengujian *software*

Pengujian *software* (perangkat lunak) meliputi pengujian pengiriman SMS menggunakan *shell* scripting secara manual.

7. Perhitungan *delay end-to-end* secara teori

Perhitungan *delay end-to-end* secara teori dilakukan dengan memasukkan data yang didapat dari operator GSM ke dalam rumus *delay end-to-end*.

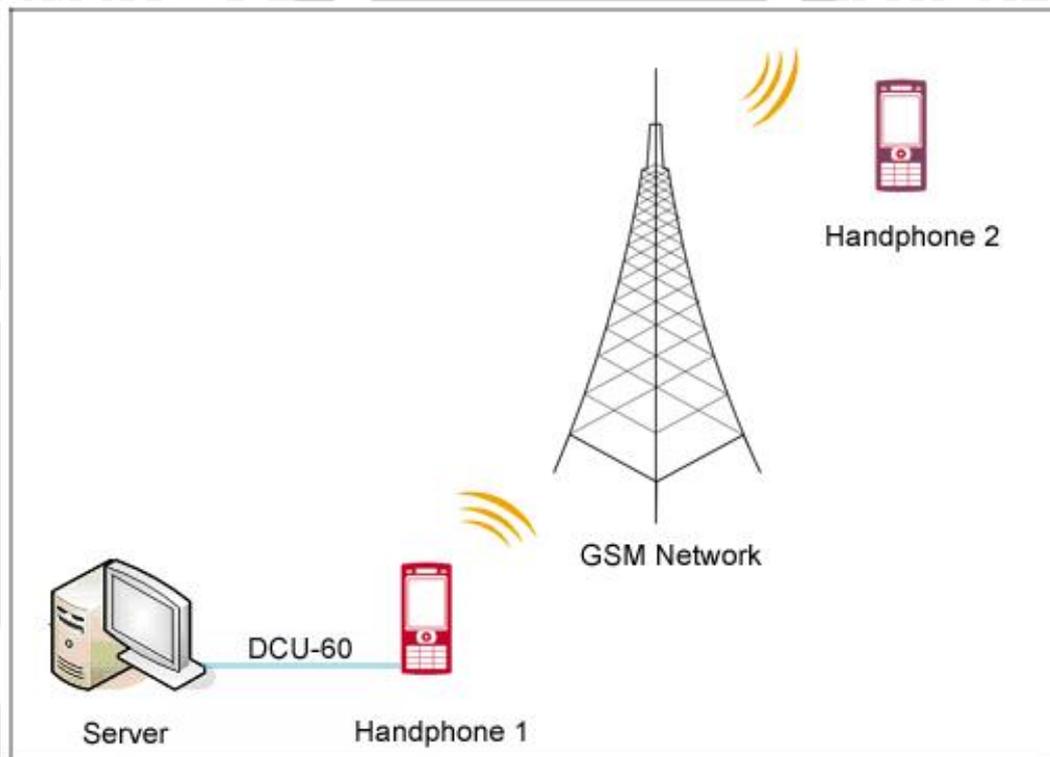
8. Perhitungan *delay end-to-end* secara praktek

Perhitungan *delay end-to-end* secara praktek dilakukan dengan menghitung dan mencatat hasil dari pengujian pengiriman sms secara manual.

BAB IV PERANCANGAN SISTEM

4.1 Perancangan Sistem

Diagram blok dari sistem monitoring koneksi *router* menggunakan SMS *gateway* seperti pada gambar 4.1:



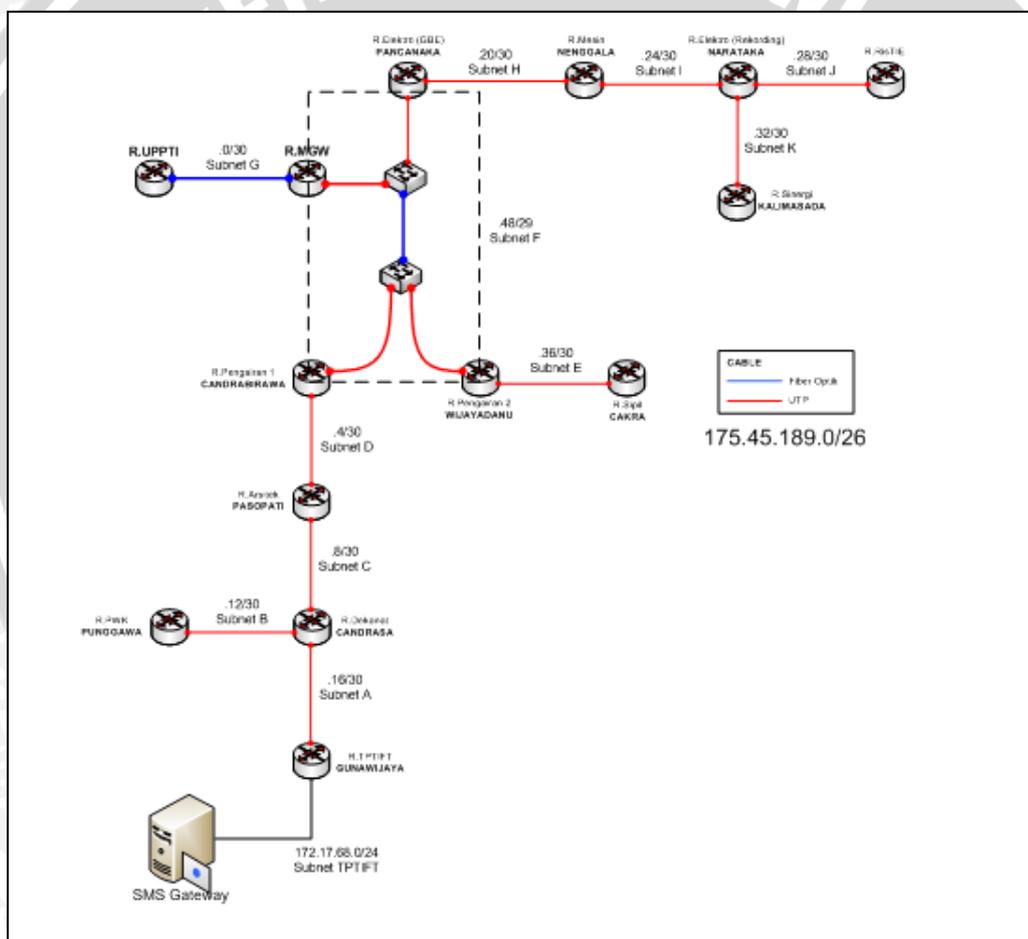
Gambar 4.1 Perancangan Sistem
[Sumber : Perancangan]

Dari gambar 4.1 dapat kita lihat :

1. *Personal Computer* (PC) yang digunakan sebagai *server* dan terintegrasi dengan program Kannel yang berfungsi sebagai SMS *gateway*.
2. Kabel data *Universal Serial Bus* (USB) DCU-60 yang berfungsi untuk menghubungkan *Handphone* (HP) dengan *Personal Computer* (PC).
3. *Handphone 1* (HP1) yakni *handphone* yang terhubung dengan *personal computer* (PC) yang berfungsi sebagai virtual SMSC (*Short*

- Message Services Center*) yang bertugas mengirimkan pesan kepada *handphone 2* (HP2) yang berfungsi sebagai *handphone* admin server.
4. *Global System For Mobile Communication Network* (GSM Network) berfungsi sebagai media pengiriman SMS dari *handphone server* ke *handphone* admin router.
 5. *Handphone 2* (HP2) sebagai *handphone* admin router yang berfungsi sebagai penerima pesan (SMS) yang telah dikirim oleh *handphone 1* (HP1).

Topologi jaringan *intranet* di Fakultas Teknik Universitas Brawijaya dapat kita lihat pada Gambar 4.2:



Gambar 4.2 Topologi Jaringan *Intranet* Fakultas Teknik Universitas Brawijaya
[Sumber : Perancangan]

4.1.1 Perangkat Keras Sistem

- Spesifikasi *Server*
 - Intel Pentium 4 cpu 2,4 GHz
 - Memory DDR 256 MB
 - Lan Card : VIA Rhine II 100Mbps full-duplex
 - Port USB (*Universal Serial Bus*)
- Spesifikasi Kabel DCU-60
 - USB *connection*
 - *Fast port system connector*
 - Data transfer
- Spesifikasi Sony Ericsson W830i
 - Jaringan EDGE,GSM 900/1800/1900
 - *Bluetooth™ technology*
 - Dukungan USB
 - Modem
 - Sinkronisasi USB

4.1.2 Perangkat Lunak

Perangkat lunak yang digunakan adalah LINUX Centos 5.3, yang digunakan sebagai *operating sistem* pada *server*. Software lain yang digunakan untuk mendukung sistem ini adalah :

- Kannel. *Software*, Kannel berfungsi sebagai SMS *gateway* dan program monitoring koneksi *router*.
- Minicom 2.1 yang berfungsi untuk menguji fungsi *AT Command* pada *handphone* dan mengetahui data PDU yang dikirim dari *handphone* pada *server* kepada *handphone* admin *router*.

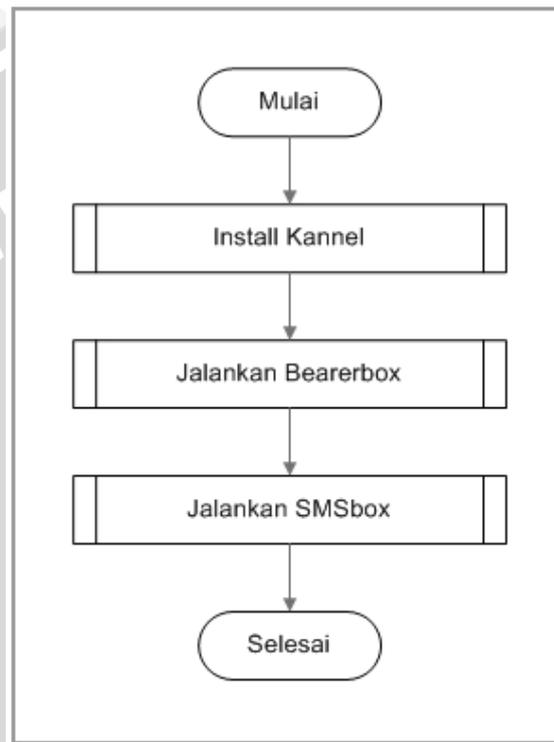
4.1.3 Prinsip Kerja Sistem

Prinsip kerja sistem monitoring koneksi *router* menggunakan SMS *gateway* adalah sebagai berikut :

1. *Server* mengirimkan perintah ping ke alamat tertentu.

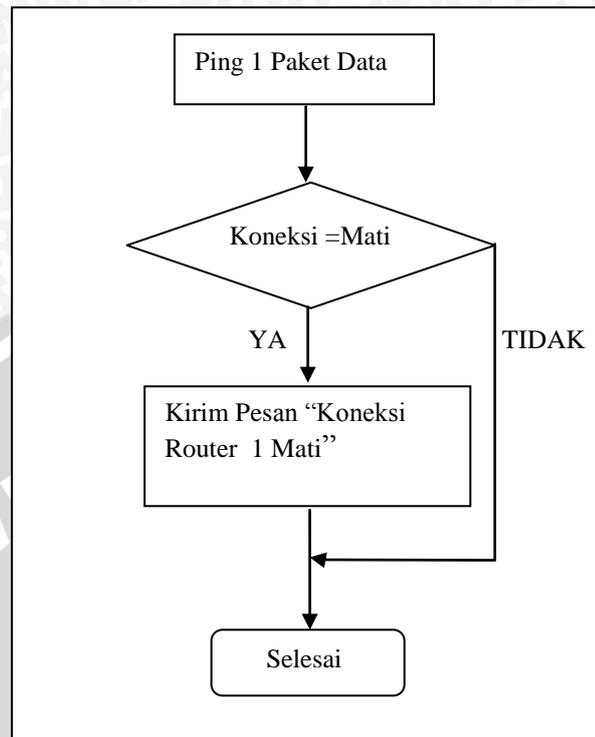
2. Dari hasil respon tersebut dapat dianalisis bahwa *router* tersebut masih tersambung dengan *intranet* atau tidak.
3. Setelah program Kannel menganalisis, maka virtual SMSC mengirimkan pesan sesuai dengan keadaan koneksi *router* kepada *handphone* admin *router*.

Pada gambar 4.3 adalah diagram alir dari SMS Gateway Kannel :



Gambar 4.3 Diagram Alir SMS Gateway Kannel
[Sumber: Perancangan]

Pada Gambar 4.4 adalah diagram alir dari program sistem monitoring koneksi router :



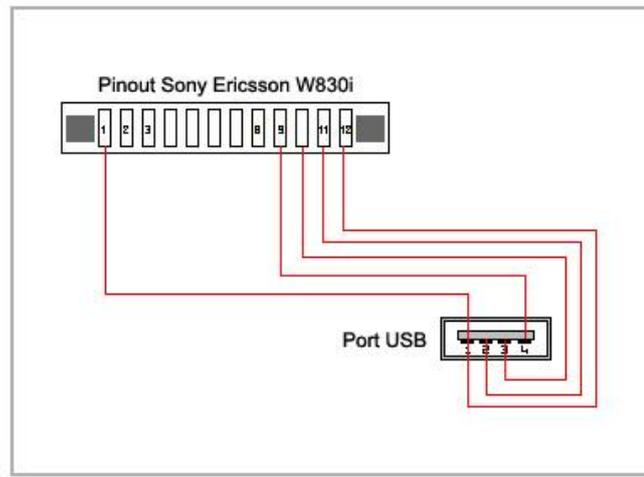
Gambar 4.4 Diagram Alir Program Sistem Monitoring Koneksi Router
[Sumber: Perancangan]

4.1.4 Rangkaian *Interface DCU-60*

Dalam perancangan ini pin-pin yang digunakan adalah sebagai berikut :

1. Pin 1 : Terhubung dengan PC USB pin 1 (VCC)
2. Pin 9 : Terhubung dengan PC USB pin 4 (GND)
3. Pin 10 : Terhubung dengan PC USB pin 3 (D+)
4. Pin 11 : Terhubung dengan PC USB pin 2 (D-)
5. Pin 12 : Terhubung dengan PC USB pin 1 (VCC)

Berikut Gambar 4.5 antarmuka *handphone* (HP) Sony Ericsson W830i dengan *port* USB pada *Personal Computer* (PC) :



Gambar 4.5 Antarmuka HP dengan Port USB pada PC
[Sumber: Perancangan]



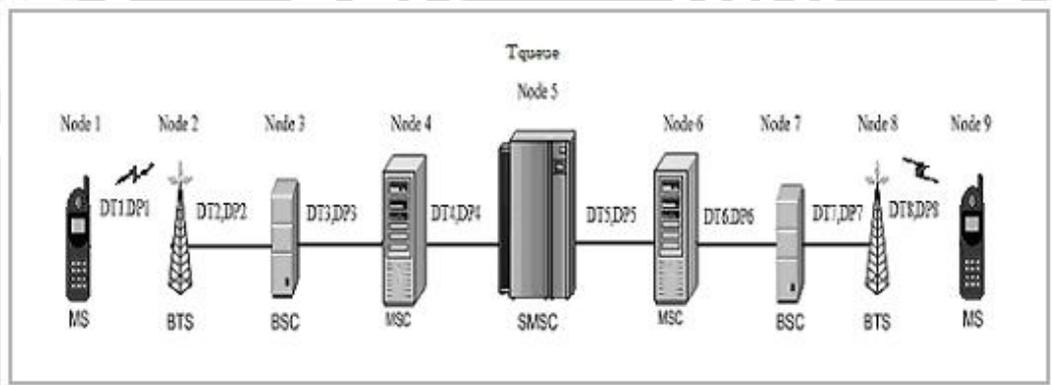
BAB V PENGUJIAN dan ANALISIS SISTEM

5.1 Perhitungan Delay

Dalam perhitungan *delay* akan dihitung waktu tunda dari SMS dikirim sampai SMS diterima oleh ponsel penerima.

Delay end-to-end

Untuk mentransmisikan *Short Message* antar jaringan GSM perlu diketahui model analisis *delay end-to-end* pada Gambar 5.1.



Gambar 5.1 *Delay end-to-end* SMS pada jaringan GSM.

[Sumber : Perancangan]

5.1.1 Perhitungan Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk mentransmisikan sebuah paket data.

Delay transmisi pada analisis ini terjadi antara MS-BTS, BTS-BSC, BSC-MSC, MSC-SMSC, SMSC-MSC, MSC-BSC, BSC-BTS, dan diantara BTS dengan MS.

Tabel 5.1 Kecepatan transmisi pada tiap *node*

| Node | Kecepatan Transmisi (Kbps) |
|----------|----------------------------|
| MS-BTS | 16 |
| BTS-BSC | 192 |
| BSC-MSC | 192 |
| MSC-SMSC | 512 |
| SMSC-MSC | 512 |
| MSC-BSC | 192 |
| BSC-BTS | 192 |
| BTS-MS | 16 |

Pada perhitungan delay ini akan dikirimkan kalimat Koneksi Router 1 Putus atau Koneksi Router 1 Hidup. Data SMS yang ditransmisikan pada kalimat tersebut adalah 22 karakter dimana 1 karakter = 7 bit, sehingga 22 karakter = 154 bit. Maka besarnya *delay* transmisi tiap *node* dapat dihitung menggunakan persamaan (2-2) yaitu :

$$DT = \frac{L}{C}$$

$$DT1 \text{ (MS-BTS)} = \frac{154 \text{ bit}}{16 \text{ Kbps}} = 9,625 \cdot 10^{-3} \text{ s}$$

$$DT2 \text{ (BTS-BSC)} = \frac{154 \text{ bit}}{192 \text{ Kbps}} = 8,02 \cdot 10^{-4} \text{ s}$$

$$DT3 \text{ (BSC-MSC)} = \frac{154 \text{ bit}}{192 \text{ Kbps}} = 8,02 \cdot 10^{-4} \text{ s}$$

$$DT4 \text{ (MSC-SMSC)} = \frac{154 \text{ bit}}{512 \text{ Kbps}} = 3,01 \cdot 10^{-4} \text{ s}$$

$$DT5 \text{ (SMSC-MSC)} = \frac{154 \text{ bit}}{512 \text{ Kbps}} = 3,01 \cdot 10^{-4} \text{ s}$$

$$DT6 \text{ (MSC-BSC)} = \frac{154 \text{ bit}}{192 \text{ Kbps}} = 8,02 \cdot 10^{-4} \text{ s}$$

$$DT7 \text{ (BSC-BTS)} = \frac{154 \text{ bit}}{192 \text{ Kbps}} = 8,02 \cdot 10^{-4} \text{ s}$$

$$DT8 \text{ (BTS-MS)} = \frac{154 \text{ bit}}{16 \text{ Kbps}} = 9,625 \cdot 10^{-3} \text{ s}$$

Sehingga *delay* transmisi total dapat diperoleh dengan menggunakan persamaan (2-3):

$$DT(\text{total}) = DT1 + DT2 + DT3 + DT4 + DT5 + DT6 + DT7 + DT8$$

$$\begin{aligned} DT(\text{total}) &= 9,625 \cdot 10^{-3} + 8,02 \cdot 10^{-4} + 8,02 \cdot 10^{-4} + 3,01 \cdot 10^{-4} + 3,01 \cdot 10^{-4} + 8,02 \cdot 10^{-4} \\ &\quad + 8,02 \cdot 10^{-4} + 9,625 \cdot 10^{-3} \\ &= 0,0231 \text{ s} \end{aligned}$$

5.1.2 Perhitungan *Delay* Propagasi

Delay propagasi adalah waktu perambatan atau penjalaran yang dibutuhkan oleh data dari suatu *node* ke *node* yang lain melalui media transmisi

Delay propagasi terjadi antara MS pengirim ke BTS dengan jarak saluran transmisi antara MS dan BTS diambil jarak terjauh yaitu 15 km dan dari BTS ke BSC dengan jarak 300 m. Kemudian antara BSC dengan MSC pengirim sejauh 10 m, antara MSC pengirim dan SMSC dengan jarak 10 m, dan dari SMSC ke MSC tujuan yang jaraknya 8 km, dari MSC ke BSC jaraknya adalah 10 m, BSC ke BTS dengan jarak 300 m, dan jarak dari BTS ke MS penerima 15 km. Maka dapat dihitung besarnya *delay* propagasi untuk layanan SMS pada jaringan GSM menggunakan persamaan (2-4) :

$$DP = \frac{d}{c}$$

$$\begin{aligned} DP1 \text{ (MS-BTS)} &= \frac{15000}{3 \cdot 10^8} \\ &= 5 \cdot 10^{-5} \text{ s} \end{aligned}$$

$$\begin{aligned} DP2 \text{ (BTS-BSC)} &= \frac{300}{3 \cdot 10^8} \\ &= 1 \cdot 10^{-6} \text{ s} \end{aligned}$$

$$\begin{aligned} DP3 \text{ (BSC-MSC)} &= \frac{10}{3 \cdot 10^8} \\ &= 3,33 \cdot 10^{-8} \text{ s} \end{aligned}$$

$$\begin{aligned} DP4 \text{ (MSC-SMSC)} &= \frac{10}{3 \cdot 10^8} \\ &= 3,33 \cdot 10^{-8} \text{ s} \end{aligned}$$

$$\begin{aligned} DP5 \text{ (SMSC-MSC)} &= \frac{8000}{3 \cdot 10^8} \\ &= 2,67 \cdot 10^{-5} \text{ s} \end{aligned}$$

$$\begin{aligned} DP6 \text{ (MSC-BSC)} &= \frac{10}{3 \cdot 10^8} \\ &= 3,33 \cdot 10^{-8} \text{ s} \end{aligned}$$

$$\begin{aligned} DP7 \text{ (BSC-BTS)} &= \frac{300}{3 \cdot 10^8} \\ &= 1 \cdot 10^{-6} \text{ s} \end{aligned}$$

$$\begin{aligned} DP8 \text{ (BTS-MS)} &= \frac{15000}{3 \cdot 10^8} \\ &= 5 \cdot 10^{-5} \text{ s} \end{aligned}$$

Sehingga besarnya *delay* propagasi total dapat diperoleh dengan menggunakan persamaan (2-5) :

$$DP(\text{total}) = DP1 + DP2 + DP3 + DP4 + DP5 + DP6 + DP7 + DP8$$

$$\begin{aligned} DP(\text{total}) &= 5.10^{-5} + 1.10^{-6} + 3,33.10^{-8} + 3,33.10^{-8} + 2,67.10^{-5} + 3,33.10^{-8} + 1.10^{-6} \\ &\quad + 5.10^{-5} \\ &= 1,288.10^{-4} \text{ s} \end{aligned}$$

5.1.3 Perhitungan *Delay* Antrian

Delay antrian adalah waktu dimana paket data berada dalam antrian untuk ditransmisikan. Selama waktu ini paket data menunggu sampai selesainya paket lain ditransmisikan.

Delay antrian total adalah penjumlahan dari waktu tunggu dalam antrian (t_{queue}) dan waktu pelayanan (t_s) dalam server. Dengan asumsi panjang *message* yang digunakan 22 karakter = 154 bit, kapasitas saluran transmisi diasumsikan yang tersedia 75% yakni sebesar 384 Kbps, dan faktor utilitas 0,9 maka dapat diketahui besarnya rata-rata kecepatan pelayanan dengan menggunakan persamaan (2-7) :

$$t_s = 1/\mu = L/C$$

$$t_s = 154 \text{ bit} / 384 \text{ Kbps}$$

$$t_s = 4,01.10^{-4} \text{ s}$$

$$t_s = \frac{1}{\mu}$$

$$\mu = \frac{1}{t_s}$$

$$= \frac{1}{4,01.10^{-4}}$$

$$\mu = 2493,76 \text{ bps}$$

Sehingga diperoleh besarnya kecepatan pesan yang datang untuk nilai kecepatan pelayanan sebesar 2493,76 bps dengan menggunakan persamaan (2-8).

$$\lambda = \rho \mu$$

Dari kecepatan pelayanan sebesar 2493,76 bps dan faktor kegunaan sebesar 0,9 (semakin besar faktor kegunaan maka semakin besar juga kecepatan pesan yang datang) maka dapat diperoleh:

$$\begin{aligned}\lambda &= \rho \cdot \mu \\ &= 0,9 \cdot 2493,76 \text{ bps} \\ &= 2244,38 \text{ bps}\end{aligned}$$

Untuk menentukan waktu tunggu pesan dalam antrian dengan menggunakan persamaan (3-9)

$$t_{\text{queue}} = \frac{\lambda}{\mu^2 (1 - \rho)}$$

dengan:

t_{queue} = rata-rata waktu tunggu dalam antrian

Dengan memasukkan nilai λ dan μ yang telah didapat, dapat ditentukan besarnya waktu tunggu rata-rata dalam antrian, yaitu:

$$\begin{aligned}t_{\text{queue}} &= \frac{\lambda}{\mu^2 (1 - \rho)} \\ &= \frac{2244,38}{2493,76^2 (1 - 0,9)} \\ &= 3,61 \cdot 10^{-3} \text{ s}\end{aligned}$$

Sehingga dengan menggunakan persamaan (2-6) dapat ditentukan *delay* antrian total yaitu :

$$\begin{aligned}T_{\text{queue}} &= t_{\text{queue}} + t_s \\ T_{\text{queue}} &= 3,61 \cdot 10^{-3} \text{ s} + 4,01 \cdot 10^{-4} \text{ s} \\ &= 40,11 \cdot 10^{-4} \text{ s} \\ &= 0,004011 \text{ s}\end{aligned}$$

Berdasarkan hasil perhitungan, diperoleh *delay* transmisi total = 0,0231 s, *delay* propagasi total = $1,288 \cdot 10^{-4}$ s, serta *delay* antrian total = 0,004011 s. Sehingga besarnya *delay end-to-end* dapat diperoleh dengan persamaan (2-1) yaitu:

$$\begin{aligned}T_{\text{end to end}} &= DT(\text{total}) + DP(\text{total}) + T_{\text{queue}} \\ &= 0,0231 \text{ s} + 1,288 \cdot 10^{-4} \text{ s} + 0,004011 \text{ s} \\ &= 0,02724 \text{ s}\end{aligned}$$

5.2 Pengaturan *baud rate*

Di dalam proses komunikasi serial antara *Personal Computer* (PC) dengan *handphone* (HP SE W830i) terlebih dahulu ditentukan *baud rate* yang digunakan. Pada sistem ini digunakan *baud rate* sebesar 115.200 bps dengan menggunakan $f_{\text{processor}} = 2700 \text{ MHz}$

Dengan rumus berikut dapat ditentukan *clock* PC yang digunakan adalah :

$$\text{Baud Rate} = \frac{f_{\text{processor}}}{16 \times (\text{clockPC} + 1)} \dots\dots\dots(5-1)$$

$$115.200 = \frac{2700 \text{ MHz}}{16 \times (\text{clockPC} + 1)}$$

$$115.200(16 \text{ clockPC} + 16) = 2700 \text{ MHz}$$

$$1843200 \text{ clockPC} + 1843200 = 2700 \text{ MHz}$$

$$1843200 \text{ clockPC} = 2698156800$$

$$\text{clockPC} = \frac{2698156800}{1843200}$$

$$\text{clockPC} = 1463,8$$

5.3 Pengujian Sistem

Pengujian alat ini terdiri dari beberapa tahap yaitu, pengujian *AT Command* dan format data SMS pada *handphone*, pengujian *software* kannel, pengujian transfer data dari *personal computer* ke *handphone* menggunakan *web browser*, pengujian *script monitoring* koneksi *router*, dan pengujian keseluruhan sistem.

5.3.1 Pengujian *AT Command* Dan Format Data SMS Pada *Handphone*

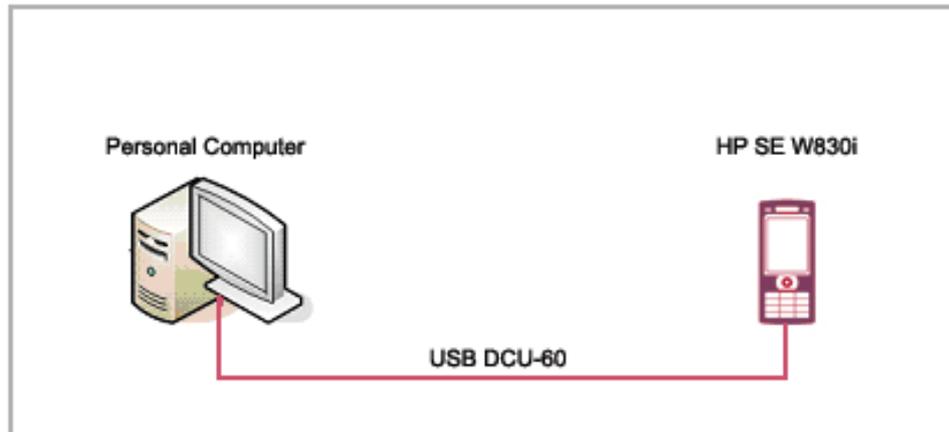
A. Tujuan

Untuk menguji fungsi *AT Command* pada *handphone* dan mengetahui data PDU yang dikirim dari *handphone* pada komputer *server* (HP1) kepada *handphone* admin *router* (HP2).

B. Peralatan yang digunakan :

1. 1 set komputer
2. *Handphone* Sony Ericsson W830i
3. Kabel data USB DCU-60

C. Blok Diagram :



Gambar 5.2 Blok diagram pengujian telepon seluler
[Sumber: Perancangan]

D. Langkah Pengujian :

- Hubungkan *handphone* dengan komputer menggunakan kabel data USB DCU-60.
- Menjalankan program Minicom
- Melakukan *setting port* USB pada program Minicom
- Mengetik instruksi AT+CMGS=20 untuk mengirim SMS
20 = jumlah pasangan heksa PDU sms dimulai setelah nomor *sms-centre* (maksimal 140)

E. Hasil Pengujian :

```

Shell - Konsole
Session Edit View Bookmarks Settings Help

Shell

Welcome to minicom 2.1
OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Sep 18 2004, 16:54:29.

Press CTRL-A Z for help on special keys

AT S7=45 S0=0 L1 V1 X4 &c1 E1 Q0
OK
AT+CMGS=20
> 0691261801000001000C91261832032441F80000A705E8329BFD06
+CMGS: 224
OK
  
```

Gambar 5.3 Hasil pengujian AT Command

[Sumber : Pengujian]

F. Analisa Hasil Pengujian :

Berdasarkan pengujian ini menunjukkan bahwa komunikasi dengan *handphone* dapat dilakukan dengan menggunakan instruksi *AT Command* dan *AT Command* merupakan bahasa yang digunakan untuk menjalankan menu-menu pada *handphone*.

Dari format data PDU yang diterima dapat dilihat ada delapan header didalamnya yaitu:

1. Nomor *sms-centre*, terdapat tiga *subheader*:
 - 06 = Jumlah pasangan heksa *sms-centre*
 - 91 = Kode internasional
 - 2618010000 = Nomor *sms-center* Telkomsel: 6281100000
2. Tipe sms:
 - Untuk *SEND* tipe *sms* = 1. Jadi bilangan heksanya adalah 01
3. Nomor referensi

Nomor referensi ini dibiarkan dulu 0, jadi bilangan heksanya adalah 00.
4. Nomor ponsel penerima, terdapat tiga *subheader*:

- 0D = Jumlah bilangan decimal nomor pengirim (13 angka)
 - 91 = Kode internasional
 - 261832032032441F8 = Nomor penerima (6281233042148)
5. Bentuk sms:
- 00 = Menandakan data dikirim sebagai sms.
6. Skema encoding:
- 00 = Menandakan skema encoding menggunakan skema 7 bit.
7. Batas waktu validitas:
- Agar sms kita pasti terkirim sampai ke ponsel penerima, sebaiknya kita tidak memberikan batasan waktu validnya.
8. Isi sms:
- 05 = Jumlah karakter dari data yang dikirim
 - E8329BFD06 = hello (data yang dikirim)

5.3.2 Pengujian Software Kannel

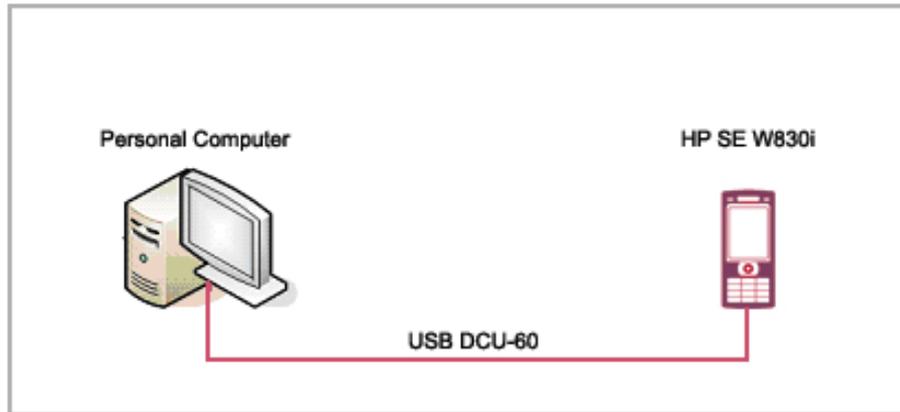
A. Tujuan :

Pengujian software kannel dilakukan untuk mengetahui apakah Personal Computer dapat berkomunikasi dengan handphone Sony Ericsson W830i.

B. Peralatan yang digunakan :

1. 1 set komputer
2. *Handphone* Sony Ericsson W830i
3. Kabel data USB DCU-60
4. Software Kannel

C. Blok Diagram :



Gambar 5.4 Blok diagram pengujian software kannel
[Sumber: Perancangan]

D. Langkah Pengujian :

- Hubungkan *handphone* dengan komputer menggunakan kabel data USB DCU-60.
- Melakukan konfigurasi kannel.conf
- Menjalankan bearerbox melalui konsole
- Menjalankan smsbox melalui konsole

E. Hasil Pengujian :

```

root@film:/etc
2010-07-19 07:19:46 [25321] [6] INFO: AT2 [GSMModem]: Logging in
2010-07-19 07:19:46 [25321] [6] INFO: AT2 [GSMModem]: init device
2010-07-19 07:19:46 [25321] [6] INFO: AT2 [GSMModem]: speed set to 115200
2010-07-19 07:19:46 [25321] [6] DEBUG: AT2 [GSMModem]: --> ATZ^M
2010-07-19 07:19:47 [25321] [6] DEBUG: AT2 [GSMModem]: <-- ATZ
2010-07-19 07:19:47 [25321] [6] DEBUG: AT2 [GSMModem]: <-- OK
2010-07-19 07:19:47 [25321] [6] DEBUG: AT2 [GSMModem]: --> AT^M
2010-07-19 07:19:48 [25321] [6] DEBUG: AT2 [GSMModem]: <-- AT
2010-07-19 07:19:48 [25321] [6] DEBUG: AT2 [GSMModem]: <-- OK
2010-07-19 07:19:48 [25321] [6] DEBUG: AT2 [GSMModem]: --> AT&F^M
2010-07-19 07:19:49 [25321] [6] DEBUG: AT2 [GSMModem]: <-- AT&F
2010-07-19 07:19:49 [25321] [6] DEBUG: AT2 [GSMModem]: <-- OK
2010-07-19 07:19:49 [25321] [6] DEBUG: AT2 [GSMModem]: --> ATE0^M
2010-07-19 07:19:50 [25321] [6] DEBUG: AT2 [GSMModem]: <-- ATE0
2010-07-19 07:19:50 [25321] [6] DEBUG: AT2 [GSMModem]: <-- OK
2010-07-19 07:19:50 [25321] [6] DEBUG: AT2 [GSMModem]: --> AT+IFC=2,2^M
2010-07-19 07:19:51 [25321] [6] DEBUG: AT2 [GSMModem]: <-- OK
2010-07-19 07:19:51 [25321] [6] DEBUG: AT2 [GSMModem]: --> AT+CPIN?^M
2010-07-19 07:19:52 [25321] [6] DEBUG: AT2 [GSMModem]: <-- +CPIN: READY
2010-07-19 07:19:52 [25321] [6] DEBUG: AT2 [GSMModem]: <-- OK
2010-07-19 07:19:52 [25321] [6] DEBUG: AT2 [GSMModem]: --> AT+CSCA="+62855000000"
^M
2010-07-19 07:19:53 [25321] [6] DEBUG: AT2 [GSMModem]: <-- OK
2010-07-19 07:19:53 [25321] [6] DEBUG: AT2 [GSMModem]: --> AT+CMGF=0^M

```

Gambar 5.5 Hasil pengujian bearerbox
[Sumber: Pengujian]

```

root@film:~
2010-07-19 07:20:43 [25378] [0] INFO: HTTP: Opening server at port 13013.
2010-07-19 07:20:43 [25378] [0] DEBUG: Started thread 1 (glib/fdset.c:poller)
2010-07-19 07:20:43 [25378] [0] DEBUG: Started thread 2 (glib/http.c:server_thread)
2010-07-19 07:20:43 [25378] [0] INFO: Set up send sms service at port 13013
2010-07-19 07:20:43 [25378] [0] DEBUG: Started thread 3 (gw/smsbox.c:sendsms_thread)
2010-07-19 07:20:43 [25378] [0] DEBUG: -----
-----
2010-07-19 07:20:43 [25378] [0] DEBUG: Kannel smsbox version 1.4.2 starting
2010-07-19 07:20:43 [25378] [1] DEBUG: Thread 1 (glib/fdset.c:poller) maps to pid 25378.
2010-07-19 07:20:43 [25378] [2] DEBUG: Thread 2 (glib/http.c:server_thread) maps to pid 25378.
2010-07-19 07:20:43 [25378] [3] DEBUG: Thread 3 (gw/smsbox.c:sendsms_thread) maps to pid 25378.
2010-07-19 07:20:43 [25378] [0] DEBUG:      dumping group (sendsms-user):
2010-07-19 07:20:43 [25378] [0] DEBUG:      <name> = <SMSThrower>
2010-07-19 07:20:43 [25378] [0] DEBUG:      <group> = <sendsms-user>
2010-07-19 07:20:43 [25378] [0] DEBUG:      <username> = <user>
2010-07-19 07:20:43 [25378] [0] DEBUG:      <concatenation> = <true>
2010-07-19 07:20:43 [25378] [0] DEBUG:      <password> = <1234>
2010-07-19 07:20:43 [25378] [0] DEBUG: Started thread 4 (gw/smsbox.c:obey_request_thread)

```

Gambar 5.6 Hasil pengujian smsbox

[Sumber: Pengujian]

F. Analisa Hasil Pengujian :

Berdasarkan pengujian ini menunjukkan bahwa bearerbox dan smsbox tidak menunjukkan pesan *ERROR*, sehingga program kannel dapat difungsikan. Pada Gambar 5.5 dapat kita lihat *report* bahwa *device* virtual SMSC *successfully opened*, *report* ini menandakan bahwa *Handphone* Sony Ericsson W830i dapat berkomunikasi dengan baik dengan *Personal Computer* (PC).

5.3.3 Pengujian Pengiriman SMS Menggunakan Kannel

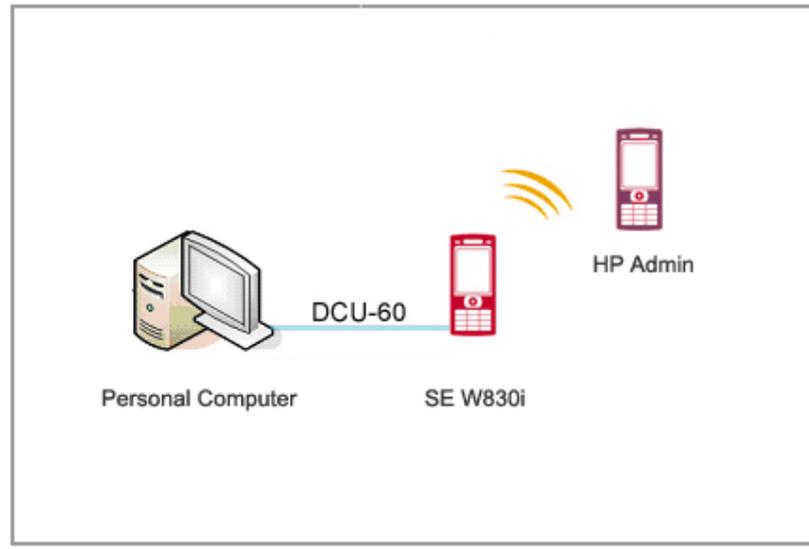
A. Tujuan :

Pengujian ini bertujuan untuk mengetahui apakah handphone Sony Ericsson W830i berfungsi dengan baik sebagai virtual SMSC.

B. Peralatan yang digunakan :

1. 1 set komputer
2. *Handphone* Sony Ericsson W830i
3. Kabel data USB DCU-60
4. Software Kannel

C. Blok Diagram :



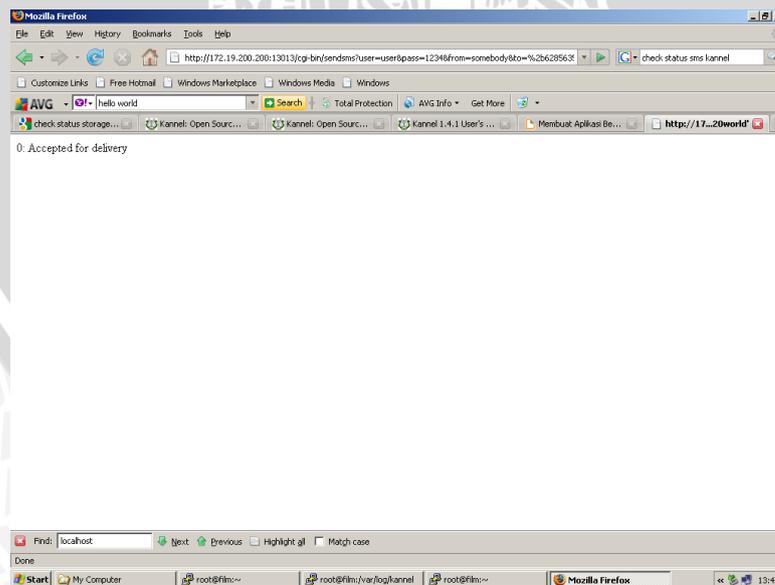
Gambar 5.7 Blok diagram pengujian pengiriman SMS

[Sumber: Perancangan]

D. Langkah Pengujian :

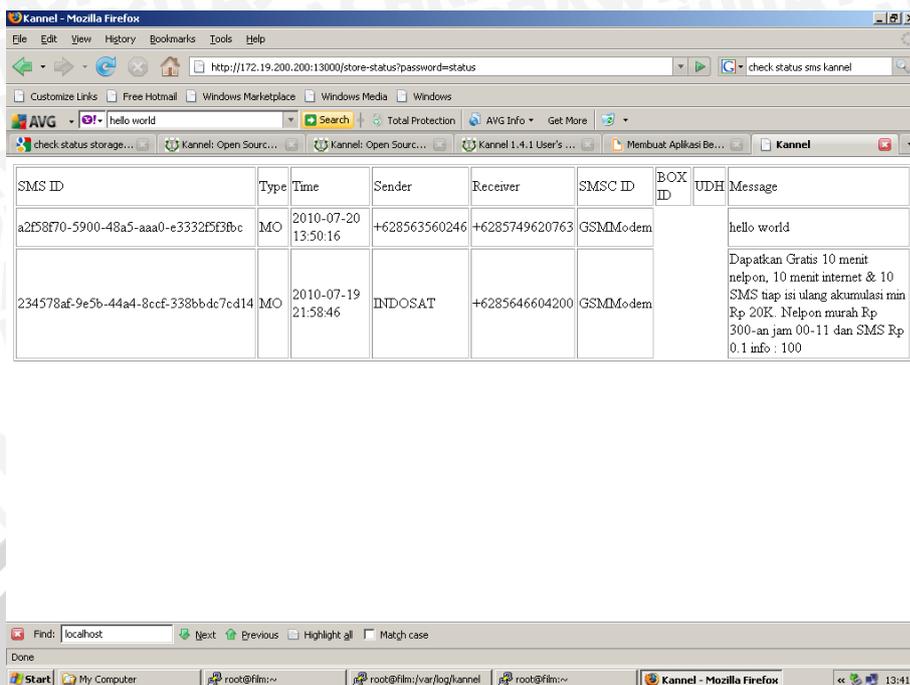
- Hubungkan *handphone* dengan komputer menggunakan kabel data USB DCU-60.
- Menjalankan *bearerbox* melalui konsole
- Menjalankan *smsbox* melalui konsole
- Mengirim pesan melalui *web browser*

E. Hasil Pengujian :



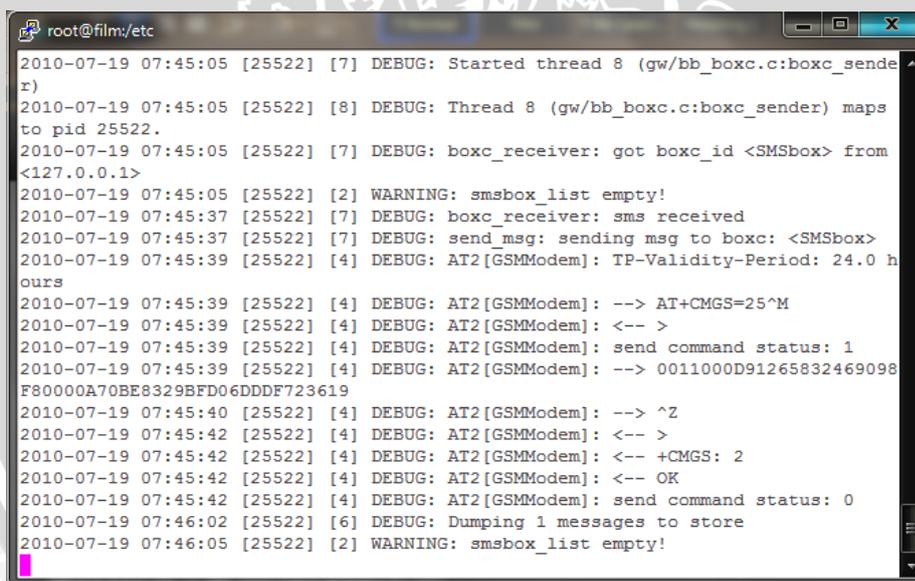
Gambar 5.8 Hasil *report* dari pengiriman SMS

[Sumber: Pengujian]



Gambar 5.9 Pengecekan status pengiriman SMS

[Sumber: Pengujian]



Gambar 5.10 Status pengiriman SMS pada bearerbox

[Sumber: Pengujian]

F. Analisa Hasil Pengujian :

- Pada gambar 5.8 terdapat *report 0: Accepted for delivery* yang berarti pesan yang telah dikirim telah diteruskan dari Personal Computer (PC) ke virtual SMSC (Handphone SE W830i) yang kemudian dikirimkan ke nomer tujuan pengiriman SMS.

Adapun kemungkinan *report* yang keluar adalah:

- 03 : *Queueing for delivery* yang berarti data yang dikirim diantrikan dan masuk ke dalam smsbox.
 - 02 : *Delivery Failure* yang berarti data yang dikirim gagal dikirimkan
 - *Authorization failed for sendsms* yang menandakan nama admin dan password yang digunakan pada saat pengiriman data salah.
- Pada gambar 5.9 terdapat :
 1. SMS ID : merupakan random karakter dari SMS yang dikirim ataupun yang diterima yang dibuat oleh software Kannel.
 2. Type : data dalam keadaan diterima atau dikirim
 3. Time : waktu pengiriman atau penerimaan pesan
 4. Sender : nomor pengirim pesan
 5. Receiver : nomor penerima pesan
 6. SMSC ID : merupakan ID dari virtual SMSC
 7. BOX ID : informasi pengiriman pesan melalui SMSbox pada Kannel
 8. UDH : identifier pesan yang masuk
 9. Message : pesan yang dikirim atau yang diterima
 - Pada gambar 5.10 dapat kita lihat bearerbox melakukan instruksi AT+CMGS=25 untuk mengirim SMS. 25 = jumlah pasangan heksa PDU sms dimulai setelah nomor *sms-centre*. Dapat kita lihat format data PDU yang dikirim adalah:
 1. Nomor ponsel penerima, terdapat tiga *subheader* :
 - 0D = Jumlah bilangan decimal nomor pengirim (13 angka)
 - 91 = Kode internasional
 - 265832469098F8 = Nomor penerima (628532469098)
 2. Bentuk sms:
 - 00 = Menandakan data dikirim sebagai sms.
 3. Skema encoding:
 - 00 = Menandakan skema encoding menggunakan skema 7 bit.
 4. Isi sms:

- 05 = Jumlah karakter dari data yang dikirim.
- E8329BFD06 DDDF723619= hello world (data yang dikirim).

5.3.4 Pengujian *Script* Monitoring Koneksi *Router*

A. Tujuan :

Pengujian ini bertujuan untuk mengetahui apakah *script* monitoring koneksi *router* yang telah dibuat dapat berjalan sesuai dengan yang diinginkan.

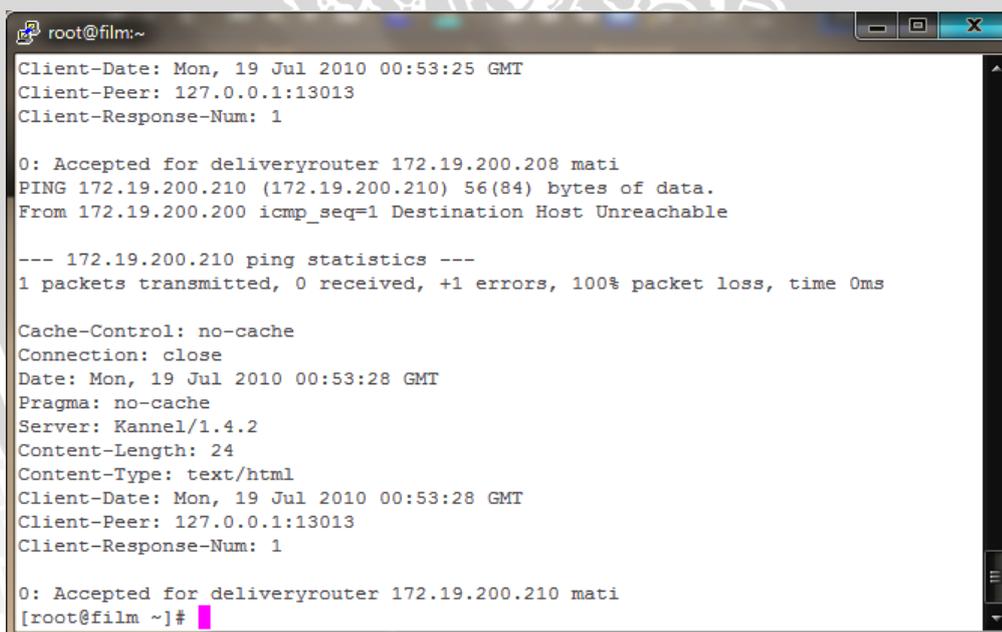
B. Peralatan yang digunakan :

- 1 set komputer

C. Langkah Pengujian :

- Menjalankan konsol pada LINUX
- Menjalankan *script ping.sh* pada console

D. Hasil Pengujian :



```

root@film:~
Client-Date: Mon, 19 Jul 2010 00:53:25 GMT
Client-Peer: 127.0.0.1:13013
Client-Response-Num: 1

0: Accepted for deliveryrouter 172.19.200.208 mati
PING 172.19.200.210 (172.19.200.210) 56(84) bytes of data.
From 172.19.200.200 icmp_seq=1 Destination Host Unreachable

--- 172.19.200.210 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

Cache-Control: no-cache
Connection: close
Date: Mon, 19 Jul 2010 00:53:28 GMT
Pragma: no-cache
Server: Kannel/1.4.2
Content-Length: 24
Content-Type: text/html
Client-Date: Mon, 19 Jul 2010 00:53:28 GMT
Client-Peer: 127.0.0.1:13013
Client-Response-Num: 1

0: Accepted for deliveryrouter 172.19.200.210 mati
[root@film ~]#

```

Gambar 5.11 Hasil pengujian *script ping.sh*

[Sumber: Pengujian]

E. Analisa Hasil Pengujian :

Pada pengecekan koneksi pada alamat IP 172.19.200.210 dapat dilihat bahwa koneksi terputus. Dari hasil pengujian ini dapat dipastikan

bahwa *script* monitoring koneksi *router* dapat berjalan sesuai dengan keadaan sesungguhnya.

5.3.5 Pengujian Keseluruhan Sistem

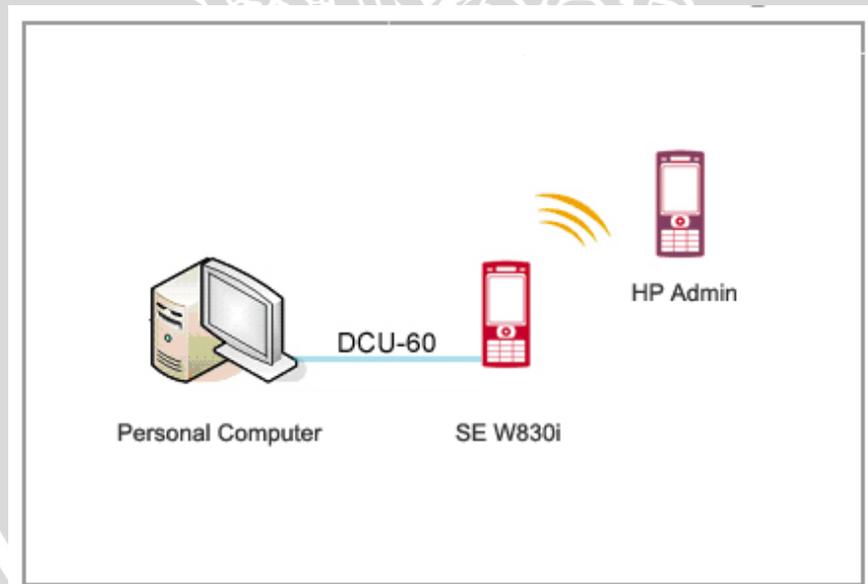
A. Tujuan :

Untuk mengetahui bahwa sistem monitoring koneksi *router* menggunakan SMS Gateway dapat beroperasi sesuai perencanaan.

B. Langkah Pengujian :

- Menjalankan *bearerbox* melalui konsole sebagai program background pada LINUX.
- Menjalankan *smsbox* melalui konsole sebagai program background pada LINUX.
- Menjalankan *script ping.sh* menggunakan *crontab* melalui konsole sebagai program background pada LINUX

Berikut gambar 5.12, blok diagram pengujian keseluruhan sistem :



Gambar 5.12 Blok diagram pengujian keseluruhan

[Sumber : Perancangan]

C. Hasil Pengujian dan Analisis :

Pengujian dilakukan dengan menjalankan *bearerbox* dan *smsbox* kannel kemudian pada layar konsole akan muncul *report AT SMSC successfully opened* yang menandakan bahwa *Personal Computer* (PC)

dengan handphone Sony Ericsson W830i (sebagai virtual SMSC) dapat berkomunikasi, dan tidak terdapat pesan *ERROR* dalam menjalankan program kannel. Selanjutnya menjalankan *script ping.sh* yang digunakan untuk memonitoring keadaan koneksi pada *Personal Computer* pada konsole LINUX dengan menggunakan crontab. Fungsi dari crontab adalah untuk mengatur jadwal jalannya *script ping.sh*. Jalannya *script ping.sh* dapat kita atur sesuai dengan waktu yang kita tentukan. Pada pengujian ini *script ping.sh* akan dijalankan setiap 1 jam. Sehingga setiap 1 jam admin *router* akan mendapatkan status keadaan koneksi *router* putus atau hidup.

5.3.6 Pengujian Waktu Tunda Pengiriman SMS

A. Tujuan :

Untuk mengetahui waktu tunda dari SMS mulai dikirim oleh server sampai SMS diterima oleh *handphone* admin.

B. Langkah Pengujian :

- Hubungkan *handphone* dengan komputer menggunakan kabel data USB DCU-60.
- Menjalankan *bearerbox* melalui konsole
- Menjalankan *smsbox* melalui konsole
- Menjalankan *script ping.sh* melalui konsole

C. Hasil Pengujian :

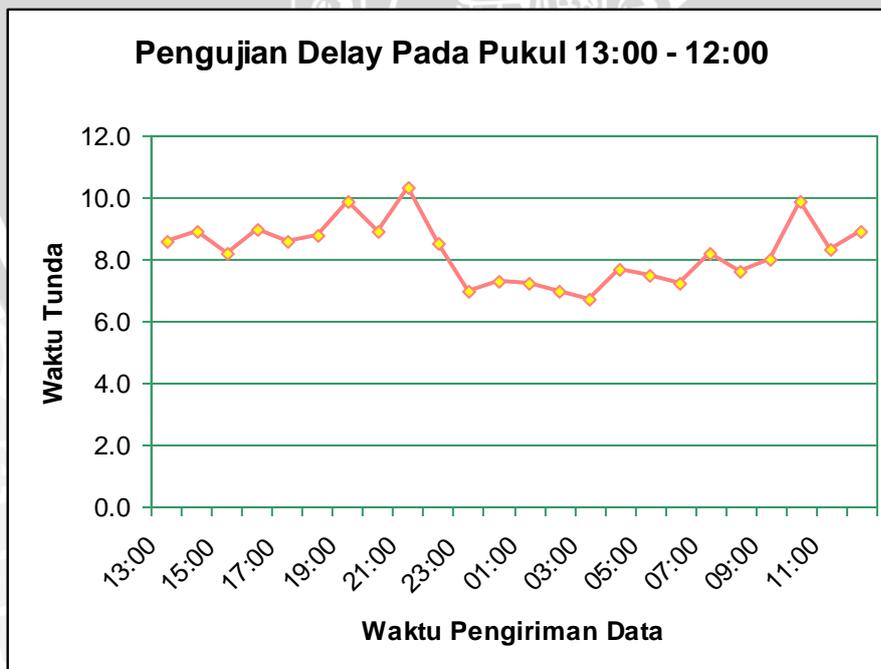
▪ Pengujian 1

Setelah dilakukan pengiriman data sebanyak 24 kali mulai pukul 13.00-12.00 didapat hasil seperti pada tabel 5.2 dan gambar 5.13.

Tabel 5.2 Hasil pengujian delay mulai pukul 13.00-12.00

| No | Jam | Waktu (s) |
|----|-------|-----------|
| 1 | 13:00 | 8.6 |
| 2 | 14:00 | 8.9 |
| 3 | 15:00 | 8.2 |
| 4 | 16:00 | 9.0 |
| 5 | 17:00 | 8.6 |
| 6 | 18:00 | 8.8 |
| 7 | 19:00 | 9.9 |
| 8 | 20:00 | 8.9 |
| 9 | 21:00 | 10.3 |
| 10 | 22:00 | 8.5 |
| 11 | 23:00 | 7.0 |
| 12 | 00:00 | 7.3 |
| 13 | 01:00 | 7.2 |
| 14 | 02:00 | 7.0 |
| 15 | 03:00 | 6.7 |
| 16 | 04:00 | 7.7 |
| 17 | 05:00 | 7.5 |
| 18 | 06:00 | 7.2 |
| 19 | 07:00 | 8.2 |
| 20 | 08:00 | 7.6 |
| 21 | 09:00 | 8.0 |
| 22 | 10:00 | 9.9 |
| 23 | 11:00 | 8.3 |
| 24 | 12:00 | 8.9 |

[Sumber: Pengujian]



Gambar 5.13 Grafik delay pengiriman data dari pukul 13.00-12.00.

[Sumber : Pengujian]



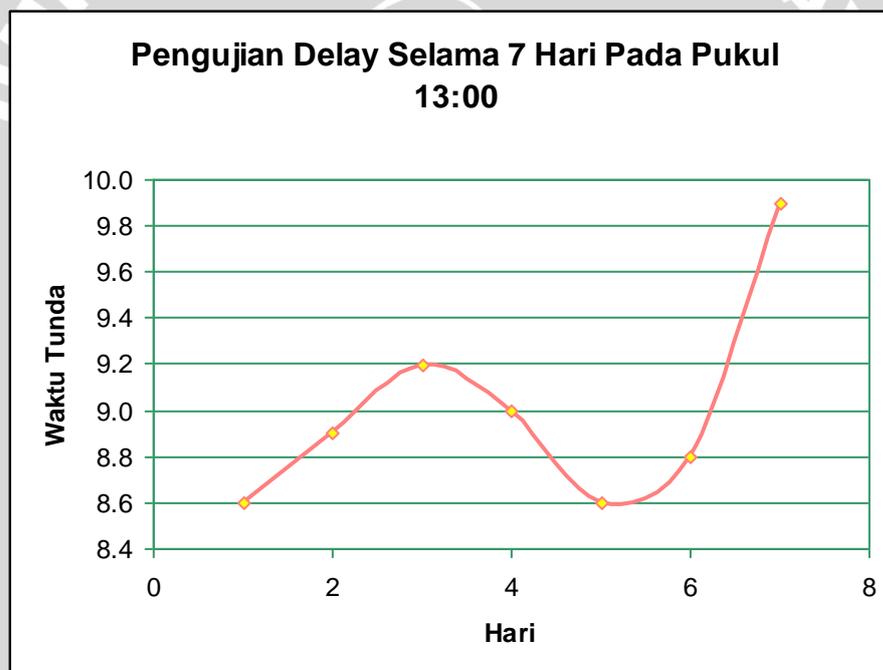
▪ **Pengujian 2**

Setelah dilakukan pengiriman data selama 7 hari setiap pukul 13.00 didapat hasil seperti pada tabel 5.3 dan gambar 5.14.

Tabel 5.3 Hasil pengujian *delay* selama 7 hari setiap pukul 13.00

| No | Jam | Waktu (s) |
|----|-------|-----------|
| 1 | 13:00 | 8.6 |
| 2 | 13:00 | 8.9 |
| 3 | 13:00 | 9.2 |
| 4 | 13:00 | 9.0 |
| 5 | 13:00 | 8.6 |
| 6 | 13:00 | 8.8 |
| 7 | 13:00 | 9.9 |

[Sumber: Pengujian]



Gambar 5.14 Grafik *delay* pengiriman data dalam 7 hari setiap pukul 13.00
[Sumber : Pengujian]

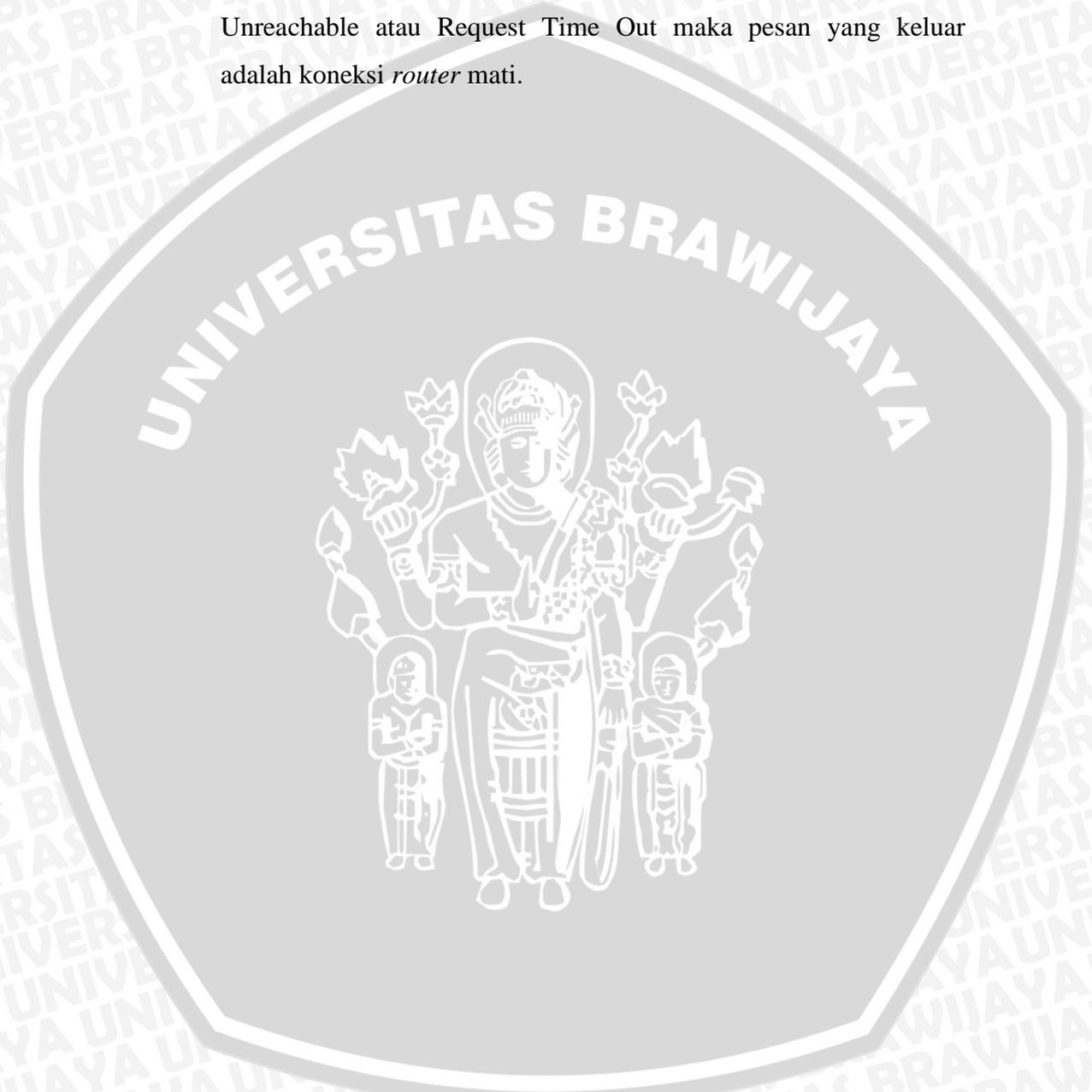
D. Analisa Hasil Pengujian :

Dari hasil pengujian 1 didapat waktu rata-rata sebesar 8,2 detik. Dari hasil pengujian 2 didapat waktu rata-rata sebesar 9 detik. Dan hal ini memang jauh lebih besar dari perhitungan yang didapat sebesar 0,02724 detik. Ini dikarenakan banyak faktor diantaranya tingkat prioritas dalam penggunaan kanal pelayanan dari jaringan GSM (*Global System for*

Mobile Communication) itu sendiri dan proses eksekusi pengiriman SMS melalui kannel.

Pada proses pemeriksaan koneksi, proses analisis program berjalan sesuai dengan keadaan koneksi yang sesungguhnya, yaitu :

- Apabila hasil dari eksekusi *script ping.sh* adalah Destination Host Unreachable atau Request Time Out maka pesan yang keluar adalah koneksi *router* mati.



BAB VI PENUTUP

6.1 Kesimpulan

Dari hasil perancangan, pembuatan dan pengujian sistem monitoring koneksi *router* menggunakan SMS *gateway* secara keseluruhan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Dari hasil pengujian *software* Kannel didapatkan *report device* virtual SMSC *successfully opened*, hal ini menunjukkan bahwa virtual SMSC dapat berjalan dengan baik.
2. Dari hasil pengujian pengiriman SMS melalui *web browser* didapatkan hasil bahwa pesan yang dikirim telah terkirim dengan report 0: *Accepted for delivery* yang menandakan pesan yang telah dikirim telah diteruskan dari Personal Computer (PC) ke virtual SMSC (Handphone SE W830i) yang kemudian dikirimkan ke nomer tujuan pengiriman SMS.
3. Dari hasil pengujian *script ping.sh* sebagai *script* monitoring koneksi *router*, apabila didapatkan *report Destination Host Unreachable* atau *Request Time Out* maka pesan yang keluar adalah koneksi *router* mati. Hasil tersebut menunjukkan bahwa *script ping.sh* telah berfungsi sesuai dengan keadaan koneksi *router* yang sesungguhnya.
4. Dari hasil pengujian didapat *delay* rata-rata sebesar 8,2 detik, hal ini memang jauh lebih besar dari perhitungan yang didapat sebesar 0,02724 detik. Ini dikarenakan banyak faktor diantaranya tingkat prioritas dalam penggunaan kanal pelayanan dari jaringan GSM (*Global System for Mobile Communication*) itu sendiri dan proses eksekusi pengiriman SMS melalui Kannel.

6.2 Saran

Ada beberapa hal yang perlu diperhatikan dalam pengembangan sistem ini dikemudian hari. Meskipun sistem ini sudah dapat bekerja dengan baik dan sesuai dengan yang direncanakan masih ada hal-hal yang perlu ditingkatkan, diantaranya:

1. Kerja sistem dapat dikembangkan untuk memonitoring keadaan jaringan mulai dari *router* sampai client.
2. Sistem monitoring koneksi *router* ini sebaiknya dirancang agar admin *router* dapat meminta data yang diinginkan, seperti data tentang keadaan trafik jaringan.



DAFTAR PUSTAKA

- Khang, Bustam. 2002. *Trik Pemrograman Aplikasi Berbasis SMS*. Jakarta: Elex Media Komputindo Kelompok Gramedia
- Purbo, Onno W., 2001. "TCP/IP". PT Elex Media Komputindo
- Susanto, Budi. 2004. *Pemrograman Script Pada UNIX/LINUX*
- Steven, W Richard, 2001. *TCP/IP Illustrated , The Protocol, Volume 2: The Implementation*
- Hunt, Craig, 2002. *TCP/IP Network Administration, 3rd Edition*
- Stallings, William. 2000. *Data & Computer Communication*. New Jersey : Prentice-Hall Inc.
- http://elektronika-elektronika.blogspot.com/2007_05_01_archive.html, Januari 2010
- <http://www.kannel.org>, November 2009
- <http://www.pinouts.ru>, November 2009
- <http://www.total.or.id>, Januari 2010
- <http://gsmserver.com/pinout/ericsson/ericst2x.php>, November 2009
- <http://www.sonyericsson.com/cws/home?cc=id&lc=id>, Januari 2010



LAMPIRAN



Konfigurasi Kannel

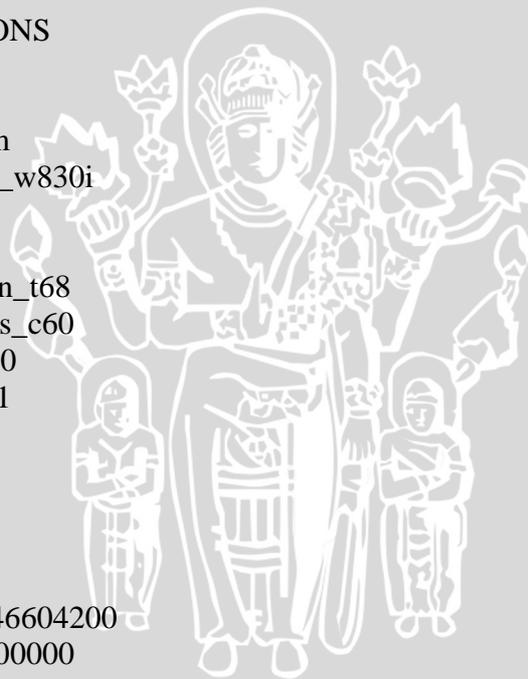
```
## sms.conf
group = core
admin-port = 13000
smsbox-port = 13001
admin-password = admin
status-password = status
#admin-deny-ip = ""
#admin-allow-ip = ""
log-file = "/var/log/kannel/bearerbox.log"
log-level = 2
#log-level = 0
access-log = "/var/log/kannel/bearerbox_access.log"
store-file = "/var/log/kannel/kannel.store"
box-allow-ip = "127.0.0.1;192.168.*.*"
#box-deny-ip = ".*.*.*"
```

SMSC CONNECTIONS

```
group = smsc
smsc = at
smsc-id = GSMModem
modemtype = ericsson_w830i
#modemtype = iTegno
#modemtype = ME45
#modemtype = ericsson_t68
#modemtype = siemens_c60
#device = /dev/ttyUSB0
device = /dev/ttyACM1
#device = /dev/ttyS0
speed = 0
#speed = 115200
#speed = 96000
#sim-buffering = true
my-number = +6285646604200
sms-center = +62855000000
#validityperiod = 168
```

modem.conf

```
group = modems
id = ericsson_w830i
#id = iTegno
need-sleep = TRUE
#need-sleep = false
#speed = 115200
need-sleep = true
sendline-sleep = 1000
#sendline-sleep = 10
message-storage = "MT"
```



```
#message-storage = "SM"  
detect-string = "Ericsson W830i"  
init-string = "AT+CNMI=2,3,2,0,0;+CMGF=0"  
#enable-mms = true  
#no-pin = true  
#no-smsc = true  
keepalive-cmd = "AT+CBC;+CSQ;+CMGF=0"  
#keepalive-cmd = "AT+CBC;+CSQ"  
#broken = true  
#no-smsc = true  
speed = 0  
#speed = 115200  
enable-hwhs = "AT+IFC=2,2"
```

```
## smsbox.conf  
group = smsbox  
bearerbox-host = localhost  
smsbox-id = SMSbox  
sendsms-port = 13013  
global-sender = 13013  
#sendsms-chars = "0123456789 +-"  
log-file = "/var/log/kannel/smsbox.log"  
log-level = 0  
#log-level = 2  
access-log = "/var/log/kannel/smsbox_access.log"
```

```
## sendsms.conf  
group = sendsms-user  
username = user  
password = 1234  
name = SMSThrower  
concatenation = true  
#max-messages = 99  
#user-deny-ip = ".*.*.*.*"  
#user-allow-ip = "127.0.0.1;192.168.*.*"  
#faked-sender = "+62817309405"
```

```
## sms service  
group = sms-service  
keyword = nop  
text = "no service"
```

Script Monitoring Koneksi Router

```
#!/bin/bash
ruter[1]=172.19.200.221
ruter[2]=172.19.200.208
ruter[3]=172.19.200.210
i=1;
while [ $i -le 3 ];
do
    case "$i" in
        "1" )
            ping -c 1 ${ruter[$i]};
            if [ $? -eq 1 ]
            then
                #zenity --info --text "Failed to reach google, panic!"
                GET -e 'http://localhost:13013/cgi-
bin/sendsms?user=user&pass=1234&from=somebody&to=%2b628563560246&t
ext=Koneksi%20router%20satu%20Mati'
                echo "router ${ruter[$i]} mati"
            fi
            ;;
        "2" )
            ping -c 1 ${ruter[$i]};
            if [ $? -eq 1 ]
            then
                #zenity --info --text "Failed to reach google, panic!"
                GET -e 'http://localhost:13013/cgi-
bin/sendsms?user=user&pass=1234&from=somebody&to=%2b628563560246&t
ext=Koneksi%20router%20dua%20Mati'
                echo "router ${ruter[$i]} mati"
            fi
            ;;
        "3" )
            ping -c 1 ${ruter[$i]};
            if [ $? -eq 1 ]
            then
                #zenity --info --text "Failed to reach google, panic!"
                GET -e 'http://localhost:13013/cgi-
bin/sendsms?user=user&pass=1234&from=somebody&to=%2b628563560246&t
ext=Koneksi%20router%20tiga%20Mati'
                echo "router ${ruter[$i]} mati"
            fi
            ;;
    esac
done
```

```
esac  
let i=$i+1;  
done
```

