PERANCANGAN PROGRAM ANALISIS KEMUNGKINAN JENIS PAKET DATA TEKS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

FERDIAN ADIPTA P.

NIM. 0510633032-63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2012

KATA PENGANTAR

Syukur alhamdulillah saya haturkan kehadirat Allah SWT, karena dengan rahmat dan hidayah-Nyalah sehingga saya dapat menyelesaikan laporan skripsi yang berjudul " Perancangan Program Analisis Kemungkinan Paket Data Teks " dengan lancar dan tepat pada waktunya.

Dalam kesempatan ini tidak lupa saya ucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan banyak bantuan kepada saya, sehingga dapat terselesaikanya penelitian saya ini. Saya berharap apa yang saya tuliskan dalam laporan ini dapat berguna bagi pembaca, teman-teman mahasiswa dan tentunya buat penulis sendiri.

Tidak lupa dalam kesempatan kali ini penulis ingin memberikan ucapan terimaksih kepada :

1. Orang tua dan saudara-saudara saya yang selalu memberikan motivasi untuk penelitian ini.
2. Muhammad Aswin, Ir., MT., dan R.Arief Setyawan,ST., MT., selaku dosen pembimbing skripsi.
3. Rekan-rekan seperjuangan, Lastono Risman Hidayat dan Bima Aditya M.S.
4. Dan pihak-pihak lain yang tidak bias disebutkan satu persatu.

Akhir kata jika ada kesalahan baik di sengaja maupun tidak disengaja selama saya melaksanakan penelitian ini, saya memohon maaf sebesar-besarnya.

Malang, 11 Juli 2012

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR GRAFIK

No Judul Halaman

ABSTRAKSI

Ferdian Adipta P. Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juli 2012, Perancangan Program Analisis Kemungkinan Jenis Paket Data Teks, Dosen pembimbing : Ir. M. Aswin, MT dan R.Arief S., ST., MT.

Paket data memiliki banyak jenis, bergantung pada jenis data yang dikirimkan. Salah satu jenis data yang melalui media jaringan adalah jenis paket data teks. Paket data ini memiliki ciri-ciri yang membedakan dengan paket data lain. Paket data teks ini akan melalui mekanisme pengiriman dan penerimaan data *TCP/IP*, sehingga untuk ukuran *file* yang lebih besar, maka *file* akan dibagi menjadi beberapa paket data. Hal ini menunjukkan bahwa paket data yang dikirimkan masih memiliki ciri dari *file* data yang asli. Jenis paket data dapat dikenali dari karakter *file* yang dimilikinya. Sebuah *file* gambar dengan ekstensi *.jpeg* memiliki struktur biner yang berbeda dengan *file* audio seperti *.wav*. Jenis *file* yang umum kita kenali adalah *file* berjenis teks. *File* berjenis teks ini memiliki karakter tertentu yang membedakan dengan jenis *file* lainnya.Oleh karena itu, pada penelitian ini, penulis ingin mencoba memprediksi kemungkinan munculnya jenis *file* teks dalam paket data pada jaringan komputer melalui karakter ASCII yang ada. Nantinya penelitian ini akan bertujuan untuk merancang program yang dapat memprediksi besarnya kemungkinan jenis paket data teks. Jenis paket data teks akan diketahui dari ciri-ciri yang ada, kemudian dihitung untuk mengetahui besarnya kemungkinan.

***Kata kunci** : Paket data, karakter file, ASCII.*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan dunia teknologi informasi dewasa ini semakin pesat. Bermacam-macam jenis komputer juga telah tersedia pada era sekarang ini. Dahulu, orang menggunakan komputer secara terpisah atau *standalone*. Data komputer diolah secara manual dan tidak terhubung dengan komputer lain untuk proses komunikasi. Tetapi setelah ditemukan dan dikembangkan jaringan komputer, sekarang interaksi antar komputer bukanlah sebuah hal yang tidak mungkin. Komunikasi antar komputer ini memungkinkan terjadinya proses pertukaran data dan informasi, bahkan hingga pembagian proses juga dapat dilakukan melalui media jaringan komputer.

Jaringan komputer merupakan sebuah aspek penting dalam dunia teknologi informasi. Sebuah sistem komputer pada era sekarang tidak dapat lagi bekerja secara sendiri. Hal ini dikarenakan keterbatasan sumber daya yang dimiliki oleh komputer tersebut. Oleh karena itu, diperlukan sebuah mekanisme yang mengatur pertukaran informasi dalam jaringan.

Layaknya sebuah peraturan lalu lintas jalan raya, dalam jaringan komputer juga memiliki aturan atau protokol tertentu untuk berkomunikasi. Protokol komunikasi jaringan sekarang ini yang diterapkan hampir oleh semua pengguna komputer di dunia adalah *Transmisson Control Protocol / Internet Protocol* atau *TCP/IP*. Proses penghantaran data oleh *TCP/IP* memiliki mekanisme tersendiri. Dalam mekanisme ini terdapat proses fragmentasi dan enkapsulasi. Proses fragmentasi adalah proses pemecahan data menjadi beberapa bagian kecil untuk dikirimkan melalui media jaringan. Proses enkapsulasi adalah proses membungkus data, seperti yang kita lakukan saat kita ingin mengirimkan paket barang ke tujuan tertentu. Tentu saja paket barang tersebut akan kita bungkus dengan baik agar isi dari paket tersebut tidak rusak ditengah jalan. Data yang dihantarkan melalui media jaringan itu kemudian disebut dengan paket data.

Paket data memiliki banyak jenis, bergantung pada jenis data yang dikirimkan. Salah satu jenis data yang melalui media jaringan adalah jenis paket data teks. Paket data ini memiliki ciri-ciri yang membedakan dengan paket data

lain. Paket data teks ini biasanya selalu melalu mekanisme pengiriman dan penerimaan data *TCP/IP*, sehingga untuk ukuran *file* yang lebih besar, maka *file* akan dibagi menjadi beberapa paket data. Hal ini menunjukkan bahwa paket data yang dikirimkan masih memiliki ciri dari *file* data yang asli.

Jenis paket data dapat dikenali dari karakter *file* yang dimilikinya. Sebuah *file* gambar dengan ekstensi *.jpeg* memiliki struktur biner yang berbeda dengan *file* audio seperti *.wav*. Jenis *file* yang umum kita kenali adalah *file* berjenis teks. *File* berjenis teks ini memiliki karakter tertentu yang membedakan dengan jenis *file* lainnya. Oleh karena itu, pada penelitian ini, penulis ingin mencoba memprediksi kemungkinan munculnya jenis *file* teks dalam paket data pada jaringan komputer.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang ada, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana cara mengetahui dan merekam isi paket data yang lewat di jaringan?
2. Bagaimana analisis isi paket data untuk mendapatkan kemungkinan jenis *file* adalah teks dalam jaringan?

1.3 Batasan Masalah

Adapun beberapa batasan masalah pada penelitian ini, yaitu sebagai berikut:

1. Parameter analisis jenis paket data adalah *file* berjenis teks dengan karakter huruf vokal.
2. Jenis data yang dianalisis adalah yang melewati jaringan komputer.
3. Aturan penggunaan bahasa pada *file* teks adalah bahasa Indonesia dengan ejaan yang disempurnakan tanpa memperhatikan singkatan dan huruf kapital.
4. Penulis menggunakan sistem operasi *Ubuntu*, aplikasi *tshark* untuk menangkap paket data, dan bahasa pemrograman *C++* untuk pengolah data dan analisis.

1.4 Tujuan

Berdasarkan permasalahan yang dirumuskan, maka tujuan yang akan dicapai pada penelitian ini adalah sebagai berikut:

1. Mengetahui dan merekam isi paket data yang lewat di jaringan.
2. Menganalisis isi paket data untuk mendapatkan kemungkinan jenis *file* adalah teks dalam jaringan.

1.5 Manfaat

Adapun beberapa manfaat yang akan diperoleh melalui pengerjaan skripsi ini adalah sebagai berikut:

1. Bagi Penyusun.
   - Dapat merancang program analisis kemungkinan jenis paket data teks.
   - Dapat mengimplementasikan ilmu yang telah didapat saat perkuliahan.
   - Memperoleh pemahaman tentang kelebihan dan kekurangan sistem yang telah dibuat.
2. Bagi Pengguna.
   - Memberikan pengetahuan untuk pengembangan sistem lebih lanjut.

1.6 Sistematika Penulisan

Adapun sistematika penulisan laporan skripsi ini adalah sebagai berikut :

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika pembahasan.

BAB II Dasar Teori

Membahas teori-teori yang mendukung dalam perancangan.

BAB III Metodologi

Berisi tentang metode penelitian yang digunakan dalam perancangan dan pengujian sistem.

BAB IV Analisis Kebutuhan dan Perancangan

Membahas tentang analisis kebutuhan dari sistem dan kemudian merancang hal-hal yang berhubungan dengan analisis tersebut.

BAB V Implementasi

Mengimplementasikan hasil perancangan ke dalam algoritma-algoritma dan kode-kode program dalam bahasa pemrograman.

BAB VI Pengujian

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VII Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran.

BAB II
TINJAUAN PUSTAKA

2.1 Teori Dasar Karakteristik dan Analisis Data Teks

2.1.1 Jenis file teks plain (*unformatted text*)

File teks adalah data dalam bentuk karakter. Data dalam bentuk karakter ini direpresentasikan dalam *binary code* dan *hexadecimal code*. Teks dalam hal ini adalah kode ASCII dan ektensi ASCII seperti *unicode* murni. Tiap karakter direpresentasikan oleh 7 *binary digit* (desimal = 0-127). Contoh *plain* text adalah saat kita mengetik dengan *notepad*. [WPD-05]

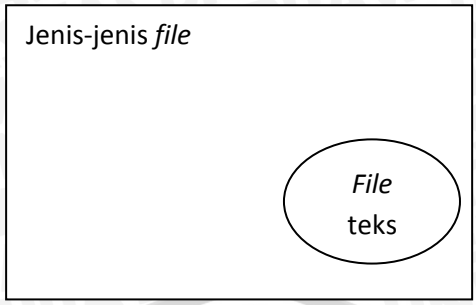
File teks dapat dikatakan ini tidak terenskripsi dan tidak mengandung *embedded information*, seperti *font*, *link*, dan *inline image*. Dalam *file* teks, terdapat perbedaan format *plain text* di *windows* dan *unix*. Di *windows*, akhir baris ditandai dengan *carriage return/CR+ Line feed/LF(\13\10).* Sedangkan di *Unix* ditandai dengan *line feed/LF (\10)* saja. [WPD-05]

2.1.2 Karakteristik jenis *file text*

Untuk menganalisis paket data yang lewat dalam jaringan, maka diperlukan sebuah mekanisme yang dimulai dari pembacaan isi paket data hingga *output* dari program analisis tersebut. Parameter analisis yang akan digunakan dalam penelitian kali ini adalah karakter huruf vokal a,i,u,e,o dalam format *hexadecimal*.

Mekanisme pembacaan paket data yang lewat dapat dilakukan dengan bantuan aplikasi *sniffing*. Aplikasi *sniffing* ini telah banyak tersedia untuk sistem operasi yang digunakan. Misal nya *Wireshark, SNORT, Tcpdump, dll*.

File teks digunakan sebagai parameter karena strukturnya yang terdiri dari karakter ASCII yang mudah dikenali. *File* teks ini merupakan salah satu dari banyak jenis *file* yang ada di dunia. Tiap karakter dalam *file* teks memiliki kode *hexadecimal* tertentu yang dapat dikenali. Oleh karena itu pengenalan karakter ini menjadi dasar pemikiran untuk penelitian kali ini. [WPD-07]

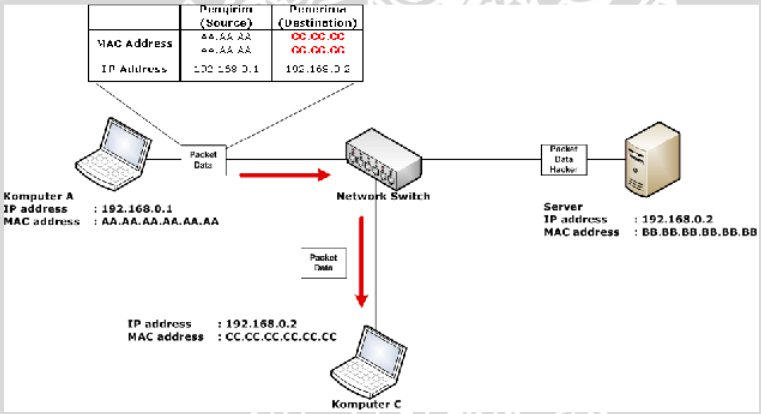


Gambar 2.1 Gambar relasi jenis *file* terhadap *file* teks.

Sumber : perancangan.

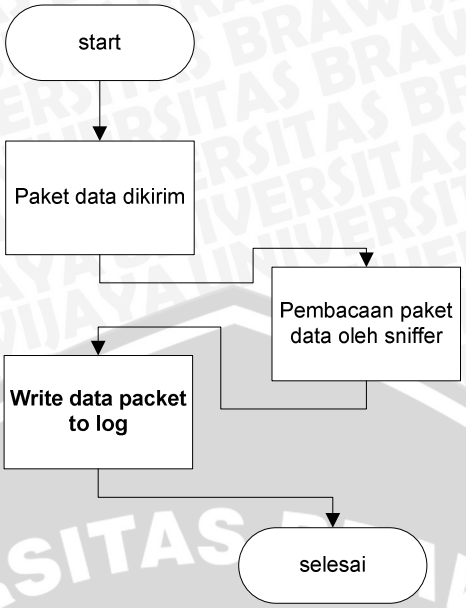
2.1.3 Mekanisme pembacaan paket data.

Paket data yang melewati jaringan perlu dibaca terlebih dahulu sebelum dapat dianalisis. Pembacaan paket data dalam penelitian ini akan meninjau lebih dalam ke arah isi dari paket data. Sedangkan bagian *header* paket data tidak diulas lebih dalam. Teknik pembacaan paket data yang digunakan adalah teknik *Sniffing*.



Gambar 2.2 Contoh topologi sederhana perekaman data jaringan.

Sumber : http://stev2711.files.wordpress.com/2011



Gambar 2.3 Mekanisme pembacaan paket data.

Sumber : http://id.wikipedia.org

2.1.4 Mekanisme analisis paket data.

Analisis paket data dalam penelitian ini menggunakan algoritma pencarian biner. Data yang dicari berbentuk karakter yang terdapat dalam paket data. Data karakter yang dicari, kemudian dihitung untuk menentukan besarnya persentase kemungkinan munculnya jenis *file* teks.

Tabel 2.1 Kode *hexadecimal* untuk karakter vokal

Karakter vokal	*Hexadecimal*
a	0061
i	0069
u	0075
e	0065
o	006F

Sumber : http: //id.wikipedia.org/wiki/ASCII

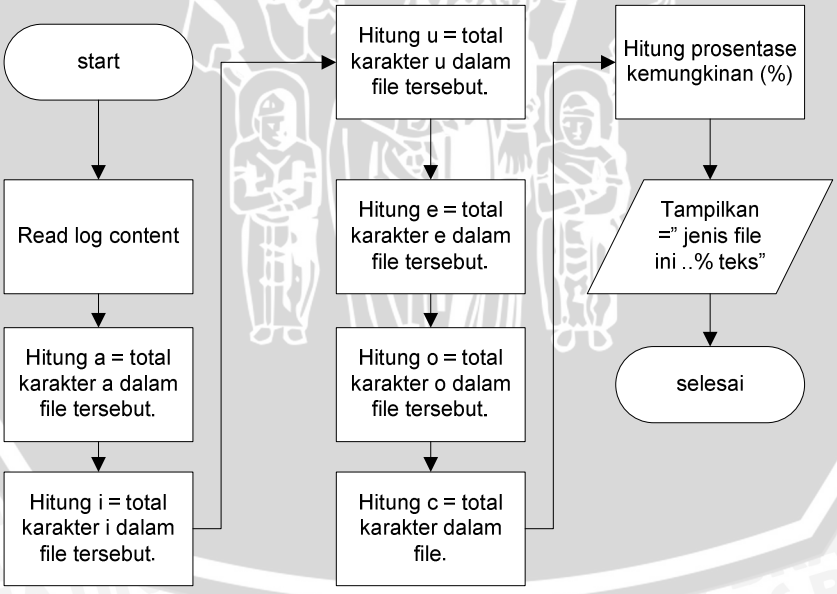
Contoh metode analisis yang digunakan dalam penelitian ini adalah sebagai berikut :

Persentase kemungkinan jenis *file* teks (%) =

$$f(n) = \frac{\sum a + \sum i + \sum u + \sum e + \sum o}{\sum c} \times 100\%$$

- *f(n)* = fungsi dari n, dimana n adalah nilai dari urutan paket data dalam log.
- $\sum a$ = jumlah total karakter huruf vokal “a”.
- $\sum i$ = jumlah total karakter huruf vokal “i”.
- $\sum u$ = jumlah total karakter huruf vokal “u”.
- $\sum e$ = jumlah total karakter huruf vokal “e”.
- $\sum o$ = jumlah total karakter huruf vokal “o”.
- $\sum c$ = jumlah total karakter dalam file tersebut.

Berikut merupakan diagram alir mekanisme analisis jenis *file*.



Gambar 2.4

Diagram alir mekanisme analisis data untuk 1 *file*.

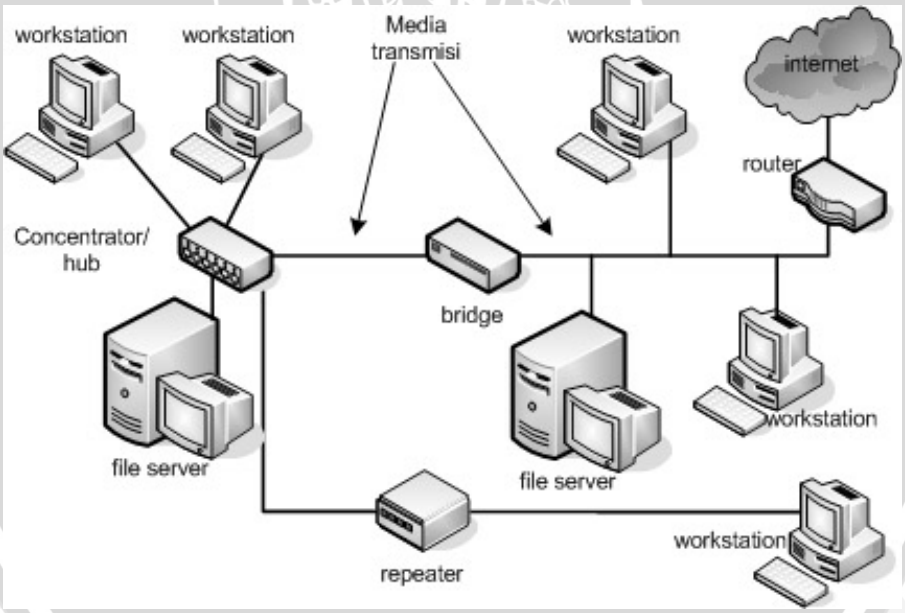
Sumber : perancangan

Melalui mekanisme analisis jenis file diatas, diharapkan tercapainya tujuan penelitian ini. Mekanisme diatas akan diulang untuk file (n) berikutnya hingga batasan (N) yang diberikan.

2.2 Teori Dasar Jaringan Komputer

2.2.1 Pengertian jaringan

Jaringan komputer adalah sebuah kumpulan komputer, *printer* dan peralatan lainnya yang saling terhubung. Informasi dan data bergerak melalui media jaringan sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada *printer* yang sama dan bersama sama menggunakan *hardware*/*software* yang terhubung dengan jaringan. Tiap komputer, printer atau *periferal* yang terhubung dengan jaringan disebut *node*. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan *node*.[WPD-04].



Gambar 2.5 Contoh jaringan komputer

Sumber : http://rezanahdi.files.wordpress.com

2.2.2 Protokol *TCP/IP*

Dalam melakukan komunikasi data, jaringan komputer menggunakan protokol-protokol komunikasi. Protokol komunikasi jaringan yang paling banyak digunakan adalah protokol *TCP/IP*. Protokol ini disusun berdasarkan konsep 7 *layer OSI(Open System Interconecction).* [STEV-94]

Protokol *TCP/IP* adalah suatu kumpulan aturan/protokol, dimana berbagai macam jenis tipe komputer, dari berbagai macam *vendor*, dengan menggunakan berbagai macam sistem operasi, dapat berkomunikasi satu sama lain. [STEV-94]

2.2.2.1 Arsitektur protokol *TCP/IP*

Protokol dalam *network*, umumnya dikembangkan dalam bentuk *layer-layer* (lapisan), dimana setiap *layer* bertanggung jawab terhadap tugas yang berbeda. Kumpulan protokol seperti *TCP/IP*, adalah kombinasi dari protokol-protokol yang berbeda, pada *layer* yang berbeda pula. *TCP/IP* terdiri dari 4 *layer*, seperti terlihat pada Tabel 2.1. [WPD-04]

Tabel 2.2 Empat *layer* dari protokol TCP/IP

Application	*Telnet, FTP, e-mail,* etc.
Transport	*TCP, UDP*
Network	*IP, ICMP, IGMP*
Link	*Device driver* dan *Interface card*

Sumber :http://www.en.wikipedia.org

1) *Link layer*, kadang kala disebut *data-link layer* atau *network interface layer*, biasanya terdiri dari *device driver* dalam sistem operasi dan *network interface card* yang terdapat di dalam komputer. Bersama-sama mereka menangani seluruh detail perangkat keras dari penghubung fisik berupa kabel (atau tipe media lain yang digunakan).
2) *Network layer*, kadang kala disebut *internet layer*, menangani pergerakan paket dalam jaringan. Berfungsi mengarahkan paket, sebagai contoh, berada

di *layer* ini. *IP* (*Internet Protocol*), *ICMP* (*Internet Control Message Protocol*), dan *IGMP* (*Internet Group Management Protocol*) berperan dalam *network layer* pada protokol *TCP/IP*.

3) *Transport layer*, menyediakan aliran data antara dua *host*, untuk *layer* aplikasi diatasnya. Pada protokol*TCP/IP* terdapat dua protokol transport yang sangat berbeda, yaitu *TCP (Transmission Control Protocol)* dan *UDP (User Datagram Protocol)*.

   - *TCP* menyediakan sebuah aliran data yang *reliable* dapat dipercaya (*reliable*) antar *host*. *TCP* berkonsentrasi pada sebuah pembagian/pemecahan data yang melewatinya dari aplikasi kedalam ukuran *chunks* (bagian data yang telah dibagi) yang tepat untuk *network layer* di bawahnya, menginformasikan paket yang diterima, menentukan *timeout* untuk membuat kepastian pemberitahuan akhir paket yang lain telah dikirim, dan seterusnya. Karena mekanisme aliran data yang dapat dipercaya telah dilakukan oleh *transport layer*, maka *application layer* dapat mengabaikan keseluruhan detail tersebut.

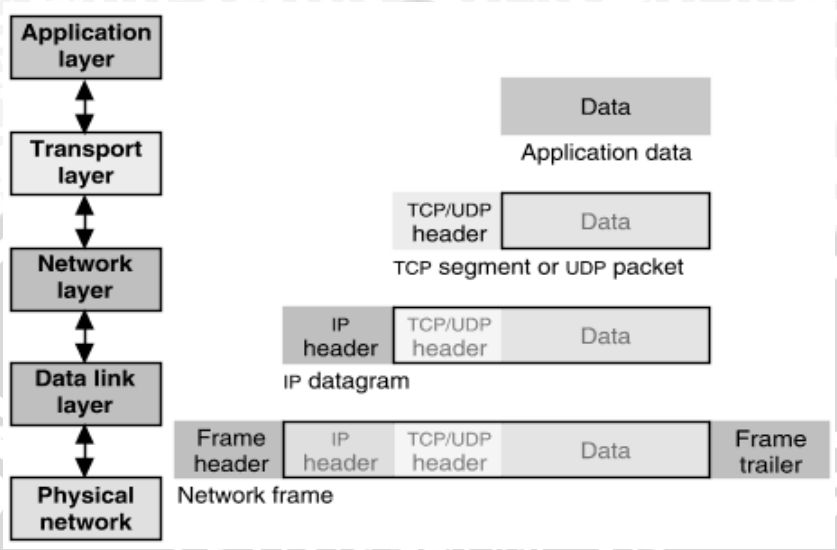
   - *UDP* disisi lain memberikan layanan yang lebih sederhana kepada *application layer*. *UDP* hanya mengirimkan paket-paket data yang disebut *datagram* dari satu *host* ke *host* lainnya, tetapi tidak ada jaminan *datagram* tersebut mencapai tujuannya. Untuk mendapatkan aliran data yang dapat dipercaya dilakukan dengan memberikan prosedur tambahan pada *application layer*.

4) *Application layer* menangani aplikasi yang ada secara detail. Terdapat banyak aplikasi *TCP/IP* yang tersedia pada setiap implementasi, antara lain:

   - *Telnet* untuk *remote login*.
   - *FTP* (*File Transfer Protocol*).
   - *SMTP* (*Simple Mail Transfer Protocol)*, untuk *electronic-mail*,
   - *SNMP* (*Simple Network Management Protocol)*.

2.2.2.2 Enkapsulasi Data

Enkapsulasi adalah proses penambahan *header* dan *ending* pada data, atau pembungkusan data. *Header* menandakan awal data, dan seringkali berisi alamat dan hal-hal lain, bergantung pada protokol dan *layer*. *Endingbit* digunakan untuk pengecekan *error*. Proses ini ditunjukkan dalam Gambar 2.4. [WPD-12]



Gambar 2.6 Proses enkapsulasi data.

Sumber : http://www.limadetik.wordpress.com

Pada setiap lapisan yang dilalui ditambahkan sebuah *header* yang berisi informasi-informasi pengontrol komunikasi. Unit data yang akan diterima pada lapisan *transport* dibagi-bagi menjadi potongan data yang disebut segmen *TCP* atau *datagram UDP* untuk aplikasi yang menggunakan *UDP* sebagai protokol *transport*nya. Segmen tersebut diteruskan ke lapisan *network* dan disebut *datagram IP*. *Datagram IP* diteruskan pada lapisan *Data Link* untuk ditransmisikan melalui media fisik jaringan. [WPD-12]

Jika suatu protokol menerima data dari protokol lain di *layer* atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu, data ini diteruskan lagi ke protokol pada *layer* di bawahnya. Proses ini disebut *encapsulation*. Hal yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada *layer* di bawahnya. Jika data ini dianggap *valid*, protokol akan melepas informasi tambahan tersebut, untuk kemudian meneruskan data itu ke protokol lain yang berada pada *layer* di atasnya. Proses ini disebut *dekapsulasi*. [WPD-12]

Setiap teknologi akses jaringan mempunyai ukuran maksimum *frame* data yang dapat ditransmisikan atau yang disebut dengan *MTU* (*Maximum Transmission Unit*). Bila ukuran *datagram IP* melebihi *MTU* maka *IP* memotong *datagram* menjadi *fragment-fragment* yang mempunyai ukuran yang tidak melebihi *MTU* jaringan. Proses pemotongan datagram menjadi *fragment* disebut fragmentasi. [WPD-12]

2.3 Teori Dasar Bahasa C.

2.3.1 Sejarah bahasa C.

Tahun 1978, Brian W. Kerninghan & Dennis M. Ritchie dari *AT & T Laboratories* mengembangkan bahasa B menjadi bahasa C. Bahasa B yang diciptakan oleh Ken Thompson sebenarnya merupakan pengembangan dari bahasa BCPL (*Basic Combined Programming Language*) yang diciptakan oleh Martin Richard. Sejak tahun 1980, bahasa C banyak digunakan pemrogram di Eropa yang sebelumnya menggunakan bahasa B dan BCPL. Dalam perkembangannya, bahasa C menjadi bahasa paling populer diantara bahasa lainnya, seperti *PASCAL*, *BASIC*, *FORTRAN*.

Tahun 1989, dunia pemrograman C mengalami peristiwa penting dengan dikeluarkannya standar bahasa C oleh *American National Standards Institute (*ANSI*)*. Bahasa C yang diciptakan Kerninghan & Ritchie kemudian dikenal dengan nama ANSI C.

Mulai awal tahun 1980, Bjarne Stroustrup dari *AT & T Bell Laboratories* mulai mengembangkan bahasa C. Pada tahun 1985, lahirlah secara resmi bahasa baru hasil pengembangan C yang dikenal dengan nama C++. Sebenarnya bahasa C++ mengalami dua tahap evolusi. C++ yang pertama, dirilis oleh *AT&T Laboratories*, dinamakan *cfront*. C++ versi kuno ini hanya berupa *compiler* yang menerjemahkan C++ menjadi bahasa C.

Pada evolusi selanjutnya, *Borland International Inc*. mengembangkan *compiler* C++ menjadi sebuah *compiler* yang mampu mengubah C++ langsung menjadi bahasa mesin (*assembly*). Sejak evolusi ini, mulai tahun 1990 C++ menjadi bahasa berorientasi obyek yang digunakan oleh sebagian besar *programmer* profesional.

Bahasa C bisa disebut bahasa pemrograman tingkat menengah (*middle level programming language*). Arti tingkat (*level*) disini adalah kemampuan mengakses fungsi-fungsi dan perintah-perintah dasar bahasa mesin/*hardware* (*machine basic instruction set)*. Semakin tinggi tingkat bahasa pemrograman maka semakin mudah bahasa pemrograman dipahami manusia, namun membawa pengaruh semakin berkurang kemampuan untuk mengakses langsung perintah dasar bahasa mesin. Demikian juga sebaliknya dengan bahasa pemrograman tingkat rendah (misalnya: *assembler*), yang semakin sulit dipahami manusia dan hanya berisi perintah untuk mengakses bahasa mesin. Dalam perspektif mudahnya dipahami manusia, C bisa digolongkan dalam bahasa tingkat tinggi, namun C juga menyediakan kemampuan yang ada pada bahasa tingkat rendah, misalnya operasi *bit*, operasi *byte*, pengaksesan memori, dsb. [GSB-11].

2.3.2 Keunggulan bahasa C.

Bahasa C pada era modern seperti sekarang ini banyak digunakan oleh *programmer* komputer dengan berbagai opini alasan seperti berikut :

1. Bahasa *C* adalah bahasa pemrograman yang paling populer saat ini, Dengan banyaknya *programmer* bahasa C, membawa pengaruh semakin mudahnya kita menemukan pemecahan masalah yang kita dapatkan ketika menulis program dalam bahasa C. Pengaruh positif lain adalah semakin banyaknya *compiler* yang dikembangkan untuk berbagai *platform* (berpengaruh ke portabilitas).
2. Bahasa *C* adalah bahasa pemrograman yang memiliki portabilitas tinggi, Program C yang kita tulis untuk satu jenis *platform*, bisa kita *compile* dan jalankan di *platform* lain dengan tanpa ataupun hanya sedikit perubahan. Ini bisa diwujudkan dengan adanya standarisasi ANSI untuk C.
3. Bahasa C adalah bahasa pemrograman yang fleksibel. Dengan menguasai bahasa C, kita bisa menulis dan mengembangkan berbagai jenis program mulai dari *operating system, word processor, graphic processor, spreadsheets*, ataupun *compiler* untuk suatu bahasa pemrograman.
4. Bahasa *C* adalah bahasa pemrograman yang bersifat moduler, Program C ditulis dalam *routine* yang biasa dipanggil dengan fungsi. Fungsi-fungsi yang telah kita buat, bisa kita gunakan kembali (*reuse*) dalam program ataupun aplikasi lain.

2.4 Teori Dasar Algoritma

2.4.1 Pengertian algoritma

Dalam matematika dan komputasi, algoritma atau *algoritme* merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma akan dapat selalu berakhir untuk semua kondisi awal yang memenuhi kriteria, dalam hal ini berbeda dengan *heuristik*. Algoritma sering mempunyai langkah pengulangan (*iterasi*) atau memerlukan keputusan (logika Boolean dan perbandingan) sampai tugasnya selesai. [ALG-11].

2.4.2 Metode dan komponen algoritma

Terdapat beberapa metode untuk merumuskan sebuah algoritma dalam program komputer, yaitu :

- Algoritma berbentuk *FlowChart* (Diagram Alir).
- Algoritma berbentuk *Pseudocode* (Kode Semu).
- Algoritma Fundamental.

Terdapat lima komponen utama dalam algoritma menurut Knuth (1973), yaitu *finiteness, definiteness, input, output,* dan *effectiveness*.

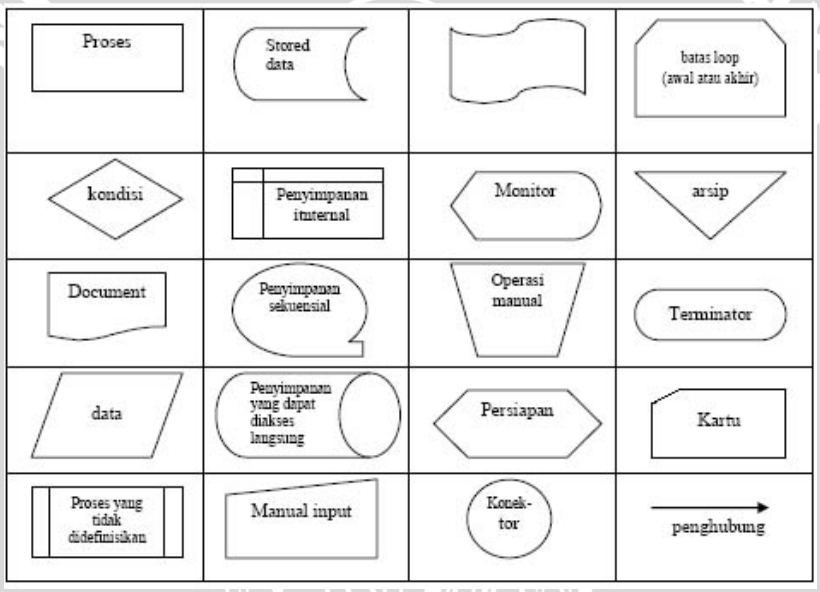
Komponen utama dalam merancang algoritma dapat dirumuskan sebagai berikut.

1. Komponen masukan, terdiri dari *variable,* jenis *variable,* tipe *variable, Constanta* dan *parameter*.
2. Komponen keluaran, yaitu merupakan tujuan dari perancangan algoritma dan program. Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran. Karakteristik keluaran yang baik adalah menjawab permasalahan dan tampilan yang ramah.
3. Komponen proses, merupakan bagian utama dan terpenting dalam merancang algoritma. Dalam bagian ini terdapat logika masalah, logika algoritma, rumusan, dan metode.[EPR-90]

2.4.3 Teori diagram alir *(flow chart)*

Diagram alir merupakan sebuah bentuk dari konsep pemrograman. Dari diagram alir inilah konsep dari sebuah program dapat dipelajari, dikoreksi, dan bahkan untuk pengembangan. Diagram alir ini memiliki simbol-simbol yang merupakan gambaran dari pernyataan logika pemrograman serta aliran logika pemrograman. Terdapat 2 jenis *flowchart*, yaitu sebagai berikut:

- Diagram alir sistem, artinya diagram alir ini menggambarkan konsep sebuah program dimana terjadi mekanisme tertentu sebagai kinerja program.
- Diagram alir proses, artinya dalam diagram alir ini menunjukan proses-proses yang terjadi dalam program, serta penjadwalan proses yang terjadi. [USU-11]



Gambar 2.7 Simbol *flowchart*

Sumber : http://www.rifan11.onsoed.ac.id

2.5 Teori Dasar Karakter ASCII (*American Standard Code for Information Interchange*).

ASCII merupakan suatu kode standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode*, tetapi ASCII lebih bersifat *universal*. Contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai dari 0000 0000 hingga 1111 1111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan Desimal. [WPD-07].

2.6 Analisis Asumsi Proporsi Karakter *File* Teks.

Dalam penelitian kali ini, penulis menggunakan teknik *sampling* untuk mengambil asumsi proporsi karakter file teks. Hal ini mengacu pada batasan masalah bahwa aturan penggunaan bahasa dan tulisan adalah bahasa Indonesia yang dengan ejaan yang disempurnakan. Perbandingan antara jumlah karakter huruf vokal dan non-vokal akan dipelajari dari berbagai contoh untuk kemudian ditarik asumsi. Sample yang diambil adalah berasal dari berbagai media seperti koran, artikel internet, dan majalah.

> **Bangkok**- Jembatan persahabatan Tahiland-Laos yang menghubungkan provinsi Nakhon-Phanom di Thailand dan Provinsi Khammouan di Laos resmi beroperasi mulai kemarin (11/11). Perdana menteri (PM) Yingluck Shinwatra menyaksikan langsung peresmian jembatan ketiga yang menghubungkan dua negara lewat jalur darat tersebut.

Sumber : Jawapos, Sabtu 12 November 2011

Analisis data :

- Jumlah karakter huruf vokal : 96 karakter.
- Total karakter : 315 karakter.
- Analisis : $t = \frac{96}{315} \times 100\% = \mathbf{30,47}\ \%$

Dalam berbagai konferensi video, wawancara dengan para jurnalis, dan pertemuan dengan para siswa di tahun-tahun terakhir ini, Maha Guru Ching Hai telah berbicara tentang keadaan darurat yang terus meningkat tentang krisis iklim di Bumi saat ini. Sebagaimana yang beliau nyatakan, "*Planet* kita adalah sebuah rumah yang sedang terbakar. Jika kita tidak bekerja sama dengan semangat kesatuan untuk memadamkan api itu, kita tidak akan memiliki rumah ini lagi." Tetapi beliau juga memberikan solusi yang mengangkat bagi umat manusia, solusi yang dengan mudah dapat dilakukan oleh setiap individu: "Jadilah *vegan* untuk menyelamatkan Bumi."

Sumber : http://www.pemanasanglobal.net/solusi/Kata-Pengantar.htm

Analisis data :

- Jumlah karakter huruf vokal : 229 karakter.
- Total karakter : 633 karakter.
- Analisis : $t = \frac{229}{633} \times 100\% = \mathbf{36,17}\ \%$

Percaya atau tidak, kalau sebanyak 73% pria mempunyai hubungan yang sangat baik dengan ibunya. Apalagi didukung dengan penelitian dari seorang pakar hubungan, psikolog Robert Van de Berg dari *Life Revolutions* yang berkata "Rasa sayang terhadap sang ibu justru bisa membantu si dia menjadi seorang penyayang yang melihat suatu hubungan dari sudut pandang positif ". Dengan begitu, keintiman dan kebebasan dalam hubungan berdua pun jadi seimbang.

Sumber : Majalah Cosmopolitan Edisi November 2010

Analisis data :

- Jumlah karakter huruf vokal : 154 karakter.
- Total karakter : 445 karakter.
- Analisis : $t = \frac{154}{445} \times 100\% = \mathbf{34,7}\ \%$

Dari data pertama didapatkan besarnya persentase sebesar 30,47 %, dari data kedua didapatkan besarnya persentase sebesar 36, 17%, sedangkan dari data ketiga didapatkan besarnya persentase sebesar 34,7 %. Maka, jarak jangkauan dari persentase yang dapat disimpulkan untuk pengambilan asumsi perbandingan karakter huruf vokal terhadap jumlah semua karakter dalam tulisan bahasa indonesia adalah pada jangkauan 30 % - 40 %.

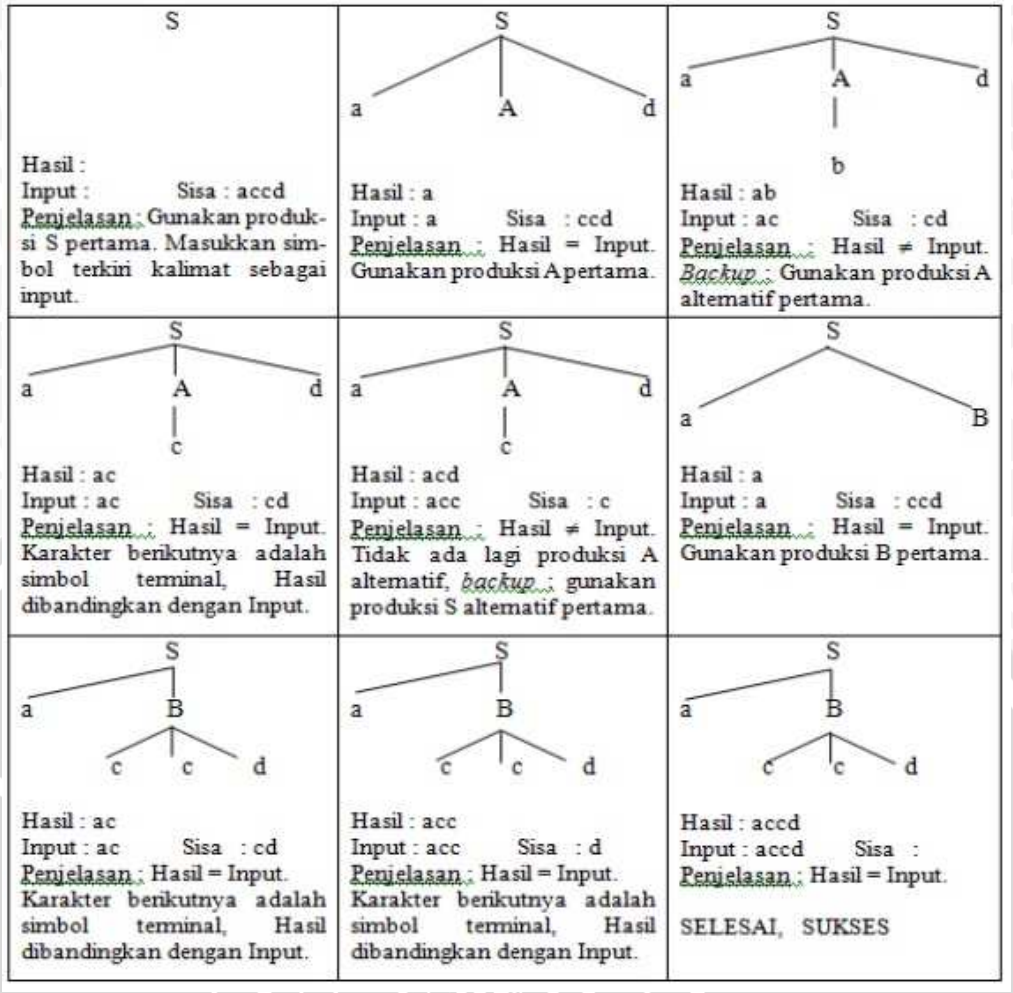
Pada penelitian kali ini, penulis akan mencoba untuk menganalisis kemungkinan jenis paket data teks, oleh karena itu materi yang akan dianalisis adalah paket data. Dalam hal ini berarti terdapat perbedaan antara format asli data seperti *text, html, ftp,* dan format dalam paket data dalam jaringan.

2.7 Teori Dasar Penguraian (*parsing*) Data

Dalam dunia ilmu pemrograman, telah dikenal istilah penguraian (*parsing*) *file* data. Definisi dari *parsing* ini sendiri adalah sebuah metode memecah suatu rangkaian masukan semisal masukan yang berasal dari *file* atau alat masukan seperti *keyboard* yang kemudian akan menghasilkan suatu pohon uraian atau *parse tree* yang akan digunakan untuk tahap berikutnya.

Metode *parsing* ini digolongkan dalam 2 jenis metode yaitu metode *parsing top-down* dan metode *parsing bottom-up*.

1. *Parsing top-down* : diberikan kalimat x sebagai masukan, *parsing* dimulai dari simbol awal S sampai kalimat X nyata (atau tidak nyata jika kalimat x memang tidak bisa durunkan dari S) dari pembacaan semua *leaf* dari pohon *parsing* jika dibaca dari kiri ke kanan.
2. *Parsing bottom-up* : diberikan kalimat X sebagai masukan. *Parsing* dimulai dari kalimat X nyata dari pembacaan semua *leaf* pohon *parsing* dari kiri ke kanan sampai tiba di simbol awal S (atau tidak sampai di S jika kalimat X memang tidak bisa diturunkan dari S). [PRS-09]



Gambar 2.8 Gambar contoh algoritma *parsing*

Sumber :

http://karmila.staff.gunadarma.ac.id/Downloads/files/1579/Scanning-Parsing+1.pdf

BAB III
METODOLOGI PENELITIAN

Dalam penelitian ini, dirancang suatu program untuk menganalisis kemungkinan jenis paket data teks menggunakan sistem operasi *Ubuntu*. Adapun metodologi yang digunakan pada penyusunan skripsi ini adalah sebagai berikut:

3.1 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan teori-teori yang mendukung dalam perencanaan dan perancangan sistem, yaitu:

1. Kajian pustaka mengenai Protokol *TCP/IP*, yaitu suatu protokol yang digunakan oleh suatu komputer dalam suatu jaringan komputer agar dapat saling berkomunikasi dengan komputer lain.
2. Kajian pustaka mengenai sistem berkas dan paket data jaringan, yaitu sebuah sistem yang mengatur tentang *file* atau berkas, serta mengatur tentang susunan, struktur, dan pengolahan paket data.
3. Kajian pustaka mengenai algoritma dan pemrograman, yaitu suatu teori yang membahas tentang algoritma yang digunakan untuk menyusun sebuah program.

3.2 Spesifikasi Alat

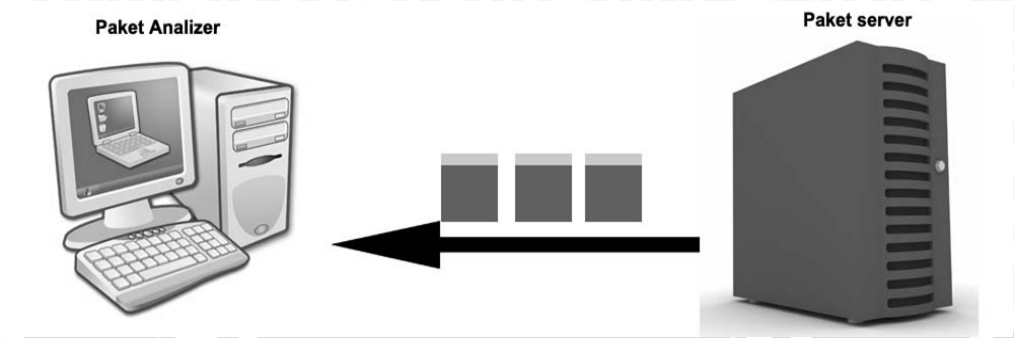
Adapun perangkat yang digunakan untuk menunjang kelancaran penyusunan skripsi ini adalah 1 unit PC sebagai penangkap paket data dan menganalisis isi paket data, dengan spesifikasi sebagai berikut:

- Processor : Intel Pentium Core Duo CPU 2,8 GHz
- Memory : 2GB RAM
- OS : Ubuntu Desktop OS.
- NIC : 1buahRealtek 10/100 Mbps Wireless LAN

3.3 Perancangan dan Implementasi Sistem.

Pada tahap ini, *interface system server* dan analisis paket data adalah sebagai berikut:

a) Perancangan topologi jaringan komputer.

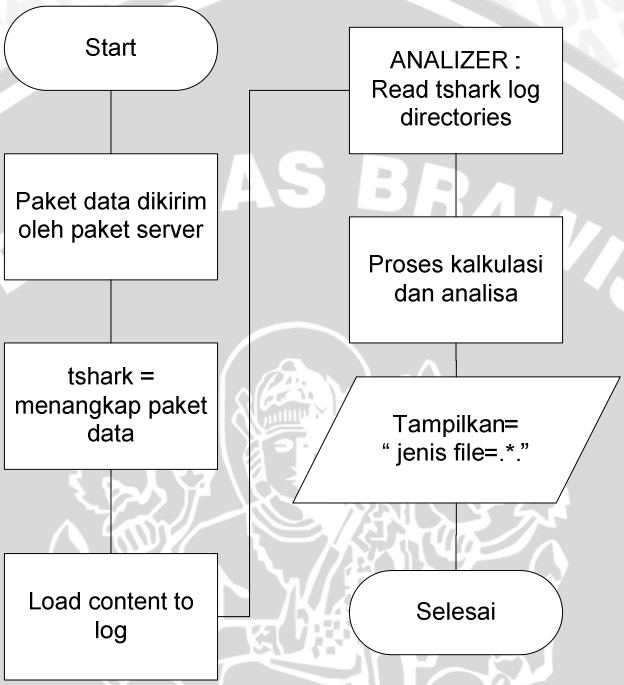


Gambar 3.1 Topologi jaringan hasil perancangan

Sumber : perancangan

b) Pemasangan (*install*) sistem operasi *Ubuntu* pada komputer *analizer*.
c) Pengkonfigurasian awal sistem operasi.
d) Pemasangan (*install*) *SNORT*dan program analisis pada *PC server* sehingga dapat bekerja sesuai harapan.

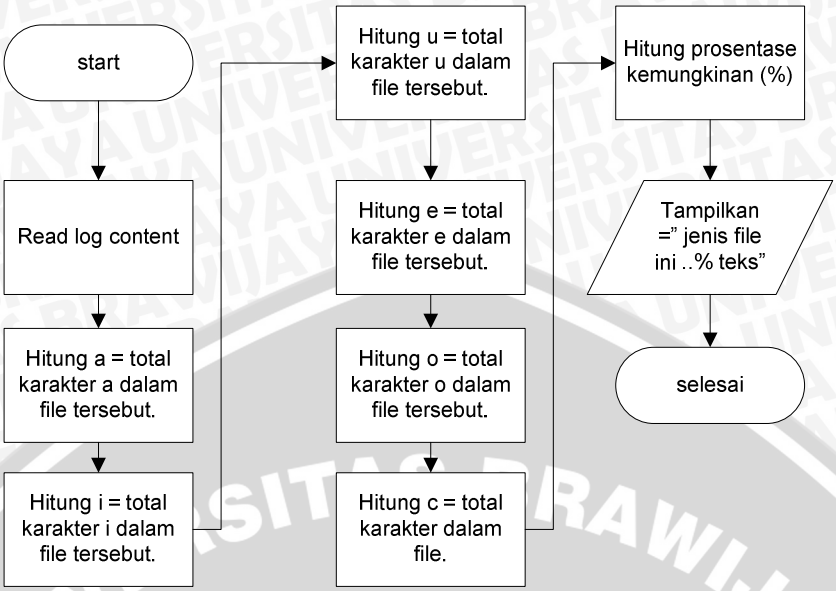
Alur kerja sistem ini dapat dilihat pada diagram alir berikut ini:



Gambar 3.2

Folwchart kerja sistem

Sumber : perancangan



Gambar 3.3

Diagram alir mekanisme analisis data untuk 1 *file*.

Sumber : perancangan

3.4 Pengambilan Kesimpulan dan Saran

Tahap ini adalah tahap pengambilan kesimpulan dari sistem yang telah dibuat. Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan dan pengujian sistem telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktek. Kesimpulan ini merupakan informasi akhir dari perancangan sistem yang berisi mengenai berhasil atau tidaknya sistem tersebut dijalankan.

Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penulisan.

BAB IV

ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

Perancangan program analisis kemungkinan jenis paket data teks ini dibagi menjadi dua tahap, yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras dalam hal ini adalah perancangan topologi jaringan komputer. Perancangan perangkat lunak meliputi perancangan jaringan komputer secara *logical*, perancangan program analisis beserta paket-paket yang mendukung, serta perancangan konfigurasi awal yang dibutuhkan.

4.1. Analisis Kebutuhan

Analisis kebutuhan dalam perancangan program analisis kemungkinan jenis paket data teks ini terdiri dari spesifikasi sistem dan cara kerja sistem.

4.1.1. Spesifikasi sistem

Spesifikasi perancangan program analisis kemungkinan jenis paket data teks adalah sebagai berikut:

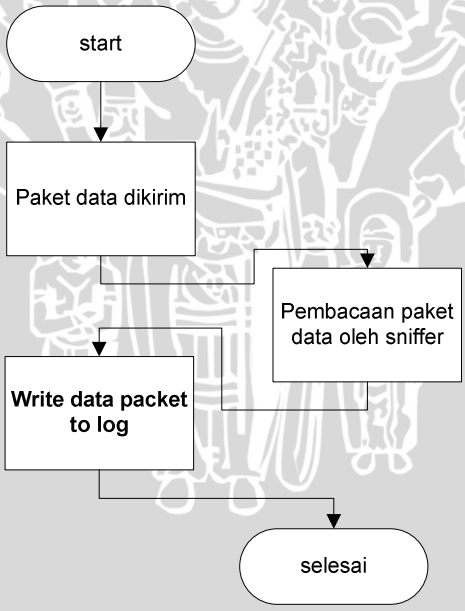
a. Terdapat 2 buah PC yang digunakan, 1 sebagai *server* dan 1 sebagai paket *analyzer*.
b. Topologi jaringan yang digunakan untuk PC *Server* dan PC *Client* adalah *peer-to-peer.*
c. PC *server* akan langsung terhubung dengan jaringan luar.
d. PC *server* akan bertugas sebagai *bridge* untuk menghubungkan jaringan luar dengan jaringan didalam, selain itu juga PC *Server* difungsikan untuk menangkap paket data yang lewat untuk direkam dan dianalisis.
e. PC *Client* akan difungsikan sebagai paket *destination* yang menerima paket data server paket data.

Untuk itu dibutuhkan seorang *user* yang bertugas sebagai *administrator* jaringan untuk mengoperasikan PC *server* tersebut. Seorang *user* yang akan mengoperasikan program analisis sesuai alur kerja program.

4.1.2. Cara kerja sistem

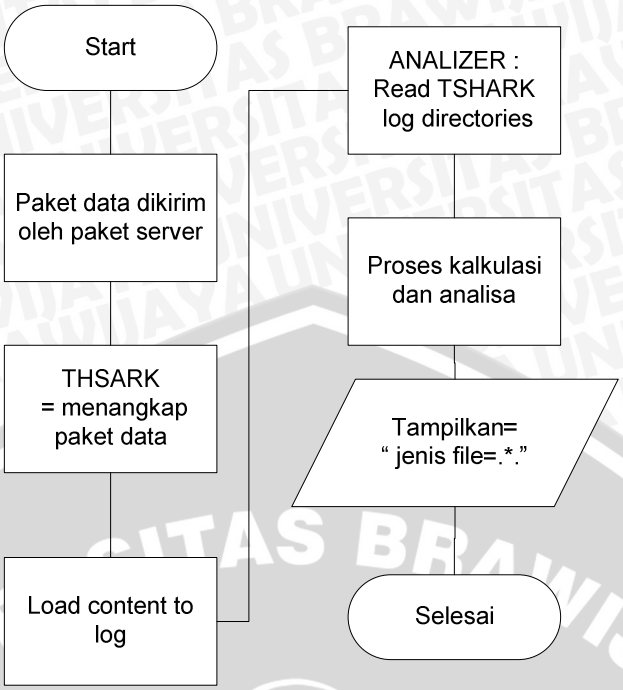
Cara kerja program analisis kemungkinan jenis paket data teks adalah sebagai berikut :

1. Paket data akan dikirimkan dari *source* menuju *server* sebagai destination. Paket data ini akan tergolong *random* dari segi jenis isi paket datanya.
2. Paket data ini akan direkam pada *server*, kemudian *server* akan membaca isi paket data tersebut.
3. Isi paket data yang telah direkam akan disalin ke dalam sebuah file di direktori lain per paket data.



Gambar 4.1 Mekanisme pembacaan paket data.

Sumber : perancangan



Gambar 4.2 Diagram alir cara kerja sistem

Sumber : perancangan

4.2. Perancangan Perangkat Keras

Perancangan perangkat keras adalah merancang suatu topologi jaringan komputer tempat sistem yang akan diimplementasikan dan juga menentukan spesifikasi *hardware* yang digunakan.

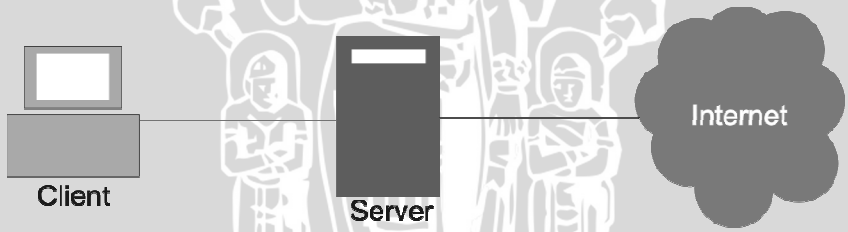
4.2.1 Spesifikasi perangkat keras.

Perangkat keras yang digunakan akan difokuskan pada PC *Server*. PC *Server* ini menggunakan sepesifikasi sebagai berikut :

- Processor : Intel Pentium Core Duo CPU 2,8 GHz
- Memory : 2GB RAM
- OS : Ubuntu Desktop OS.
- NIC : 1buah Realtek 10/100 Mbps Wireless LAN

4.2.2 Topologi jaringan yang digunakan.

Untuk mengaplikasikan sistem yang telah dirancang, maka perlu dibuat topologi sederhana seperti pada gambar 4.1 berikut.



Gambar 4.3 Topologi jaringan hasil perancangan**.**

Sumber : perancangan

4.3. Perancangan Perangkat Lunak

Perangkat lunak yang akan diimplementasikan dirancang dengan beberapa tahap secara berurutan sebagai berikut: perancangan *server* sebagai penangkap paket data dan perancangan program analisis kemungkinan jenis paket data.

4.3.1. Perancangan server untuk *sniffing* paket data.

Server sniffing adalah sebuah mesin/komputer yang berfungsi sebagai tempat dimana proses *sniffing* dilakukan. *Server* ini sekaligus juga berfungsi sebagai *gateway* atau gerbang utama bagi suatu paket data saat menuju *network* lain diluar *internal network. Server* ini sebenarnya adalah sebuah *router* yang di dalamnya juga terdapat fasilitas yang berfungsi untuk *sniffing* paket data.

Server ini menggunakan *Sistem Operasi Ubuntu*. Agar komputer ini dapat berfungsi sebagaimana mestinya, perlu dilakukan hal-hal sebagai berikut:

a. Melakukan instalasi Sistem operasi *Ubuntu*.
b. Instalasi dan konfigurasi *tshark* sebagai aplikasi *sniffing*.
c. Pengujian kinerja *sniffing*.
d. Konfigurasi *tshark* agar dapat merekam paket data dan meyimpannya dalam sebuah direktori yang ditentukan.

4.3.2. Perancangan program analisis paket data.

Program analisis paket data ini adalah sebuah program hasil perancangan yang akan menganalisis isi paket data hasil *sniffing* berdasarkan parameter yang diberikan.

Berikut merupakan analisis kebutuhan program agar dapat melakukan kerja sesuai tujuan penelitian.

a. Program dapat membaca isi paket data dalam direktori yang telah ditentukan.
b. Program dapat menghitung isi paket data berdasarkan parameter teks atau non-teks dan pendekatannya dilakukan menggunakan kode *hexadecimal* yang terdapat didalam paket data.
c. Program dapat menganalisis hasil perhitungan terhadap parameter yaitu penggunan huruf vokal terhadap total karakter yang terdapat pada tiap paket data.
d. Perhitungan program harus mengikuti rumus :

$$f(n) = \frac{\sum a + \sum i + \sum u + \sum e + \sum o}{\sum c} \times 100\%$$

- *f(n)* = fungsi dari n, dimana n adalah nilai dari urutan paket data dalam log.
- $\sum a$ = jumlah total karakter huruf vokal “a”.
- $\sum i$ = jumlah total karakter huruf vokal “i”.
- $\sum u$ = jumlah total karakter huruf vokal “u”.
- $\sum e$ = jumlah total karakter huruf vokal “e”.
- $\sum o$ = jumlah total karakter huruf vokal “o”.
- $\sum c$ = jumlah total karakter dalam *file* tersebut.

e. Hasil perhitungan baik karakter vokal ataupun jumlah total karakter merupakan pendekatan dari parameter, dan ada kemungkinan meleset dari kondisi sebenarnya.

4.3.3 Algoritma perancangan program analisis.

Perancangan program analisis paket data ini dilakukan dengan mengikuti beberapa mekanisme kerja. Mekanisme kerja tersebut bersifat *sequential* atau berurutan sehingga sifat prosesnya tidak dapat dilakukan dalam satu waktu bersamaan.

1. Mekanisme pembacaan *log file*.

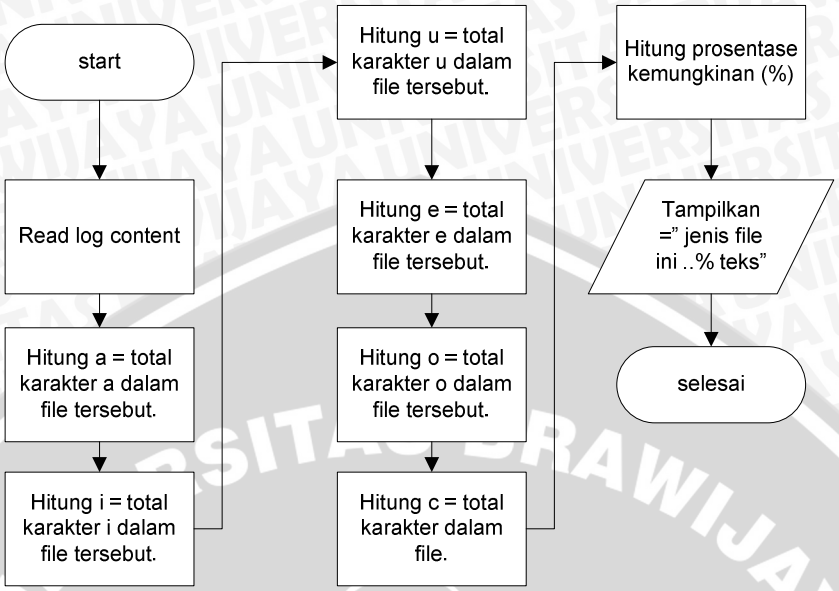
   Pembacaan file ini dilakukan pada *folder /root* tempat dimana *log file* dari program *sniffing* menempatkan *log file* hasil *export* rekaman paket data.

2. Mekanisme penghitungan dan analisis hasil rekaman.

   Hasil rekaman paket data yg tersimpan dalam *log* akan diubah yang semula berformat ekstensi *.pcap* menjadi *.txt*. Hal ini karena program analisis dirancang untuk menggunakan format *plain text*. Setelah hasil rekaman dirubah ke format *.txt*, maka kemudian akan dianalisis oleh program *sprint.exe*.

3. Mekanisme keluaran program.

   Keluaran program yang diharapkan adalah berupa tampilan persentase kemungkinan jenis teks. Keluaran akan menampilkan persentase tiap paket yang direkam. Jadi hasil tampilan adalah menunjukan persentase per-paket data terekam bukan secara akumulasi.



Gambar 4.4 Mekanisme analisis program**.**

Sumber : perancangan

4.3.4 Algoritma pengambilan asumsi program

Asumsi adalah parameter yang digunakan peneliti dalam memberikan batasan pada hasil keluaran program. Tujuan dari pemberian asumsi ini adalah untuk memberi batasan untuk keluaran program agar sesuai arah dari hasil yang diharapkan. Pengambilan asumsi program akan mengacu pada keluaran program, yang artinya asumsi ini akan sedikit berbeda dengan asumsi yang disebutkan pada bab sebelumnya.

Berikut merupakan prosedur pengambilan asumsi.

1. Peneliti melewatkan data berupa teks tertentu dalam protokol ftp, sebanyak n paket data.
2. Hasil keluaran program sebanyak n paket data akan dicatat hasilnya.
3. Mengamati nilai keluaran program dan menetukan nilai maksimal (M) dan minimal (m) hasil keluaran.
4. Mengulangi prosedur nomor 1 - 3 hingga sebanyak 3 kali.

5. Menghitung nilai rata-rata M dan m dari ketiga percobaan menggunakan persamaan :

$$\bar{M} = \frac{M1 + M2 + M3}{3}$$

$$\bar{m} = \frac{m1 + m2 + m3}{3}$$

Dimana :

- M1 = Nilai maksimum percobaan ke 1
- M2 = Nilai maksimum percobaan ke 2
- M3 = Nilai maksimum percobaan ke 3
- m1 = Nilai minimum percobaan ke 1
- m2 = Nilai minimum percobaan ke 2
- m3 = Nilai minimum percobaan ke 3
- $\bar{M}$ = Nilai rata – rata M.
- $\bar{m}$ = Nilai rata – rata m.

6. Kemudian nilai μ M dan μ m dimasukan dalam batas keluaran program.

$$\mathrm{A} = \sum_{\bar{m}}^{\bar{M}} \bar{n}$$

Dimana :

- A = Nilai asumsi terbatas.
- $\bar{n}$ = Nilai rata-rata keluaran program.

7. Nilai asumsi yang digunakan adalah nilai A. Artinya bahwa jika hasil analisis dari program untuk suatu paket data berada pada di dalam skala nilai A, maka paket data tersebut dianggap berjenis teks, dan jika berada di luar skala nilai tersebut maka paket tersebut dianggap non teks.

BAB V

IMPLEMENTASI

Program analisis kemungkinan jenis paket data teks yang telah dibahas sebelumnya akan dijabarkan dalam kesempatan kali ini. Implementasi program akan dijabarkan menjadi beberapa pokok bahasan.

5.1. Implementasi *Server* Analisis

Pada tahap ini, rancangan *server* analisis yang telah dibuat sebelumnya akan diwujudkan. *Server* analisis ini terdiri dari dua bagian besar, yaitu bagian perangkat keras dan bagian perangkat lunak.

5.1.1. Perangkat keras *server* analisis

Pada bagian perangkat keras, komputer yang digunakan memiliki spesifikasi yang bervariasi, namun secara umum adalah sebagai berikut:

- Processor : Intel Pentium Core Duo CPU 2,8 GHz
- Memory : 2GB RAM
- OS : Ubuntu Desktop OS.
- NIC : 1buahRealtek 10/100 Mbps Wireless LAN

5.1.2. Perangkat lunak *server* analisis

Pada tahap ini, perangkat keras yang telah disusun tadi akan dikonfigurasi perangkat lunaknya.

5.1.3 Konfigurasi server

Konfigurasi pada komputer *server* dilakukan setelah Sistem Operasi *Linux Ubuntu Desktop* ter-*install*. Berikut ini adalah tahap-tahap yang perlu dilakukan agar komputer *server* ini dapat berfungsi dengan baik.

a. Mengaktifkan komputer sebagai *root.*

Konfigurasi ini dilakukan dengan menambahkan *password* pada *user* "*root*". Cara ini dilakukan dengan mengetikan : " *passwd root* ". Kemudian kita *login* sebagai *root* pada *user login*.

b. Konfigurasi *ip address*.

Konfigurasi ini dilakukan dengan mengubah pengaturan pada konfigurasi *ip* dengan masukan *ip address* 99.99.99.25 *subnet* 255.255.255.0 *gateway* 99.99.99.1.

c. Memasang DNS

Konfigurasi ini dilakukan dengan menambahkan *nameServer* 202.134.0.155 dan *nameServer* 202.134.1.10 .

d. Instalasi *tshark* sebagai program penangkap paket data.

Tshark berfungsi sebagai penangkap paket data dalam jaringan. *Tshark* ini diinstall dengan mengetikan perintah :

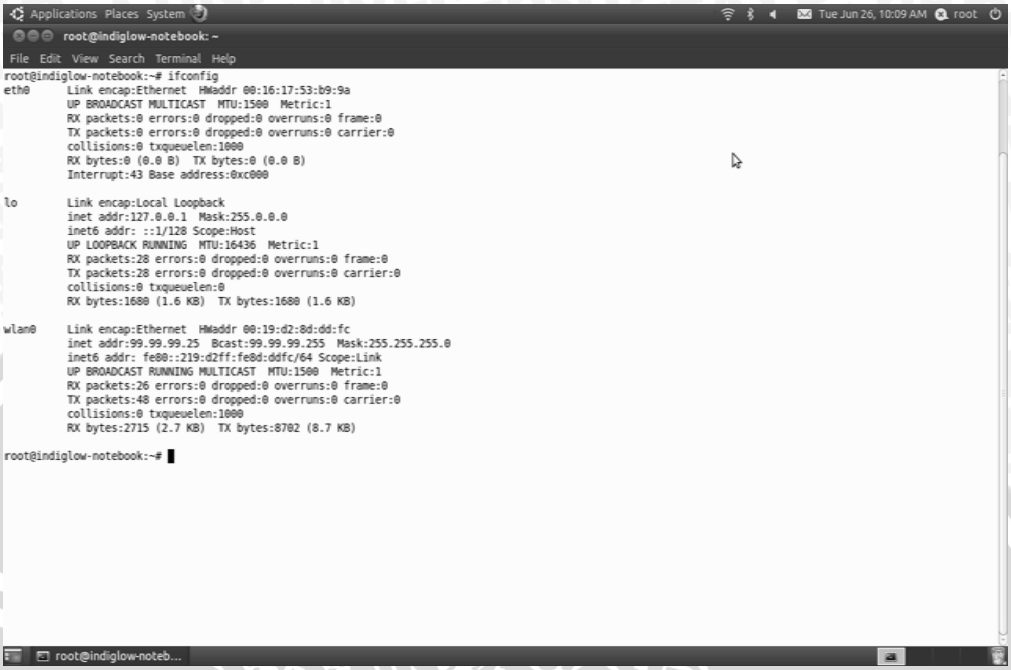
```
root@indiglow# sudo apt-get install tshark
```

e. Penggunaan Parameter *tshark*.

Parameter yang akan digunakan dalam menjalankan aplikasi *tshark* adalah :

- " –i " : Berfungsi untuk menentukan *ethernet* yang akan digunakan. Contoh : " root@indiglow # tshark –i wlan0.
- " –c" : Berfungsi untuk menentukan jumlah paket data yang direkam selama program berjalan. contoh : "root@indiglow # tshark –c 10".
- " –w" : Berfungsi untuk menuliskan hasil rekaman paket data ke suatu file. contoh : " root@indiglow # tshark –w ferdi.pcap "
- " –r" : Berfungsi untuk membaca hasil rekaman pada suatu file. Contoh : " root@indiglow# tshark –r ferdi.pcap
- " –x" : Berfungsi untuk menampilkan hasil perekaman beserta isi paket data. contoh : " root@indiglow# tshark –x ".

- " >" : Berfungsi untuk mengarahkan isi file tertentu menuju ke file tertentu. Contoh : " root@indiglow# tshark –r ferdi.pcap > ferdi.pcap.txt ".



Gambar 5.1 Gambar tampilan konfigurasi alamat ip.

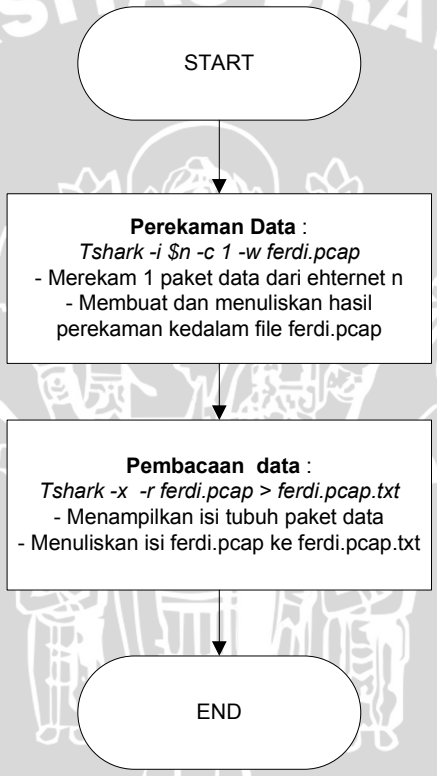
Sumber : perancangan

5.2 Implementasi Algoritma

Pada subbab 5.2 ini membahas tentang implementasi algoritma program analisis jenis paket data teks.

5.2.1 Implementasi algoritma perekaman dan penulisan paket data

Algoritma perekaman paket data dilakukan menggunakan program *tshark* yang telah terpasang. Program ini akan merekam paket data, kemudian menuliskan kedalam file *ferdi.pcap,* dan kemudian mengarahkan file *ferdi.pcap* ke sebuah file bernama *ferdi.pcap.txt*. *File* keluaran terakhir inilah yang nantinya akan diolah.



Gambar 5.2 Implementasi algoritma perekaman

Sumber : perancangan

Kerja perekaman paket data dapat dijabarkan sebagai berikut :

1. Program *tshark* akan merekam sebuah paket data pada *ethernet wlan0*, menampilkan seluruh hasil rekaman, dan menuliskan pada sebuah *file ferdi.pcap*.
2. Perintah yang digunakan untuk kerja pada poin 1 adalah :

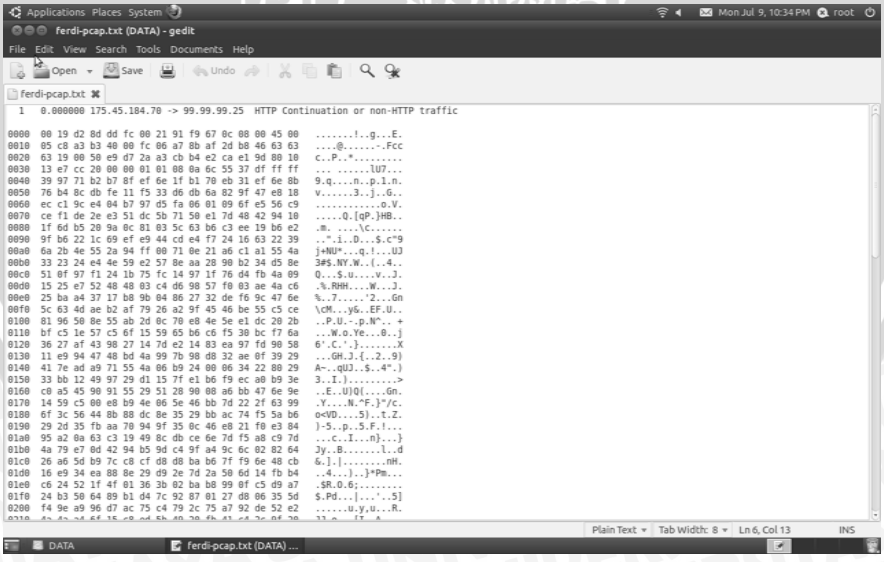
```
root@indilow# tshark –c 1  –i wlan0  –w  ferdi.pcap
```

3. Program *tshark* akan membaca hasil rekaman paket data dengan menyertakan isi paket data dari *file ferdi.pcap* dan mengarahkan ke *file ferdi.pcap.txt*.

4. Untuk kerja pada poin 3 akan menggunakan perintah :

```
root@indiglow# tshark –x –r  ferdi.pcap > ferdi.pcap.txt
```

5. Dengan demikin maka sebuah paket data telah terekam dalam *file ferdi.pcap.txt* untuk kemudian diolah.



Gambar 5.3 Gambar tampilan ferdi.pcap.txt

Sumber : perancangan

5.2.2 Implementasi algoritma analisis data.

Algoritma analisis data akan merujuk pada *file ferdi.pcap.txt* pada folder */root*. Berikut merupakan langkah kerja untuk program analisis.

1. Program akan membuka file "*ferdi.pcap.txt*" pada *folder* dimana program diletakan. Jika file tidak ditemukan maka akan menampilkan " *file could not be located"*.
2. Program akan memberikan nilai awal "0" untuk hasil pencarian karakter pada fungsi " *void reset"*.
3. Program akan memeriksa tiap karakter berdasarkan fungsi " *void checkhuruf* " dengan menggunakan karakter spasi sebagai parameter pemisah (*parsing*) file.
4. Program menghitung jumlah karakter a, i, u, e, o dalam *format heksadesimal.*
5. Program akan menghitung kemungkinan jenis paket data berdasarkan perhitungan sebagai berikut :

   Persentase kemungkinan jenis file teks (%) =

$$f(n) = \frac{\sum a + \sum i + \sum u + \sum e + \sum o}{\sum c} \times 100\%$$

   - *f(n)* = fungsi dari n, dimana n adalah nilai dari urutan paket data dalam log.
   - $\sum a$ = jumlah total karakter huruf vokal "a".
   - $\sum i$ = jumlah total karakter huruf vokal "i".
   - $\sum u$ = jumlah total karakter huruf vokal "u".
   - $\sum e$ = jumlah total karakter huruf vokal "e".
   - $\sum o$ = jumlah total karakter huruf vokal "o".
   - $\sum c$ = jumlah total karakter dalam file tersebut.
6. Program akan menampilkan hasil perhitungan dalam satuan persen (%) yang menunjukan besarnya kemungkinan jenis paket data adalah teks. Jika semakin mendekati asumsi yang diberikan, maka kemungkinan jenis paket data teks akan semakin besar.

5.2.3 Implementasi penulisan program analisis.

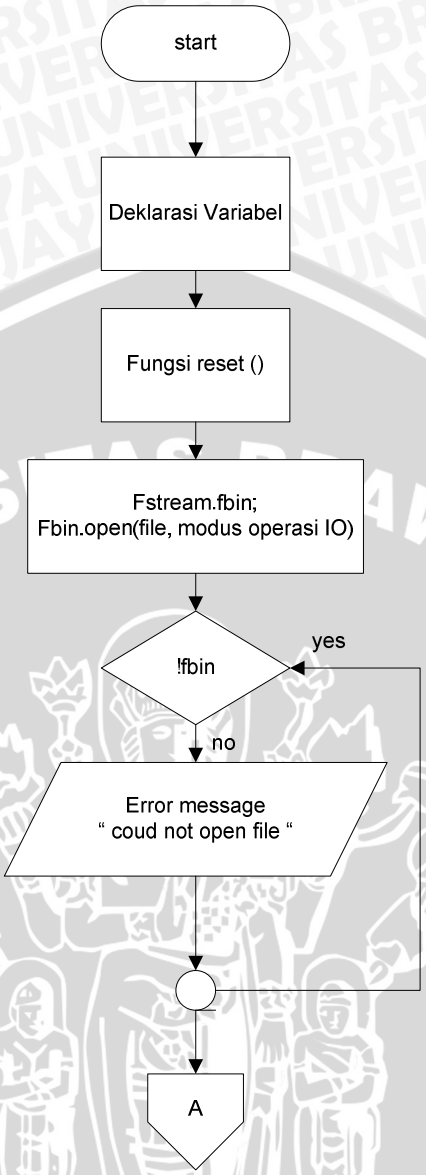
Penulisan program analisis ini menggunakan bahasa pemrograman C, program *compiler* menggunakan *g++*. Berikut merupakan *syntax* dari program analisis yang bernama *sprint.exe.*

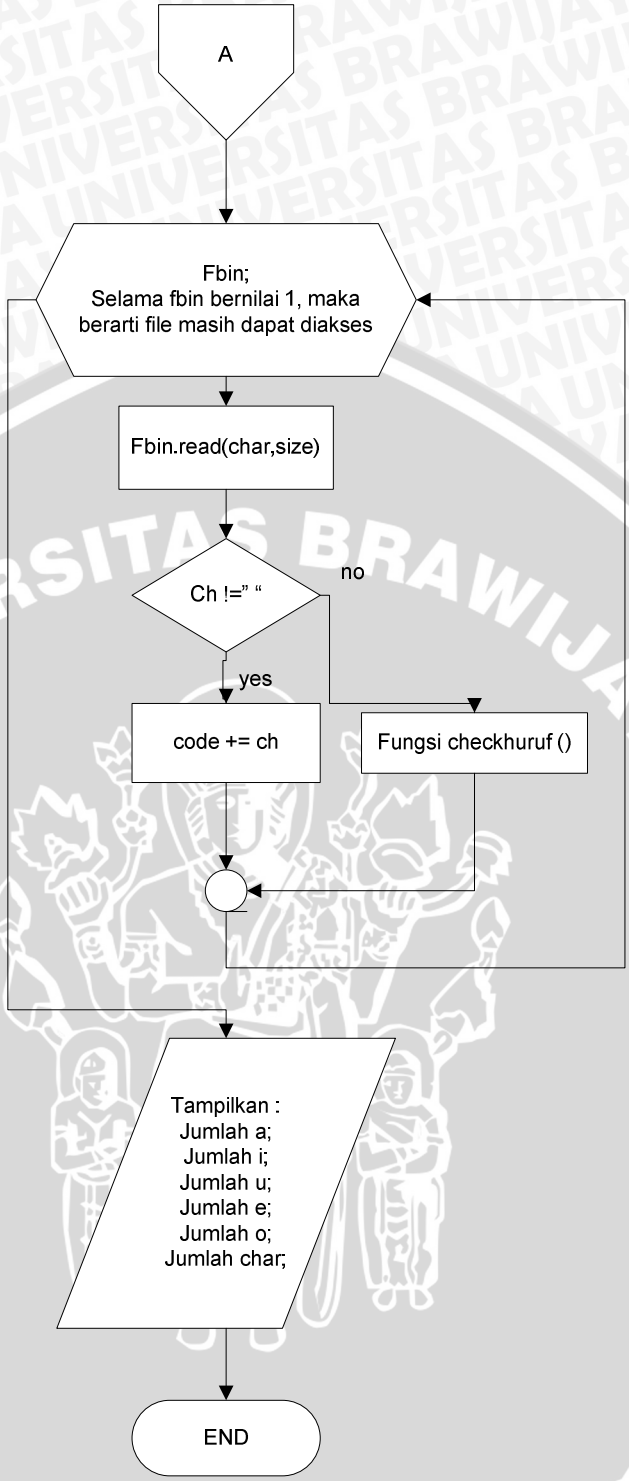
Syntax Program Sprint.exe

```
#include <fstream>
#include <string>
#include <stdlib.h>
#include <iostream>
#include <string>
using namespace std;

struct sumHuruf {
        int a;
        int i;
        int u;
        int e;
        int o;
        int charc;
}Jumlah;

void reset ();
void checkHuruf(string x);
int main()
{
        string filename;
        char ch;
        string code;
        double n;
        int i;
```

```
filename = "./ferdi.pcap.txt";
code = " ";

reset ();

fstream fbin;
fbin.open(filename.c_str(), ios::binary | ios::in | ios::out);
if (!fbin)
{
        cout << "Could not open file " << filename;
        return -1;
}
while (fbin)
{
        fbin.read((char*)&ch, sizeof (char));
        if (ch!=' '){
                code += ch;
        }
        else {
                checkHuruf(code);
                code.erase();

        }
}
cout<<"Jumlah a = "<<Jumlah.a<<endl;
cout<<"Jumlah i = "<<Jumlah.i<<endl;
cout<<"Jumlah u = "<<Jumlah.u<<endl;
cout<<"Jumlah e = "<<Jumlah.e<<endl;
cout<<"Jumlah o = "<<Jumlah.o<<endl;
cout<<"Jumlah char = "<<Jumlah.charc<<endl;

n=Jumlah.a+Jumlah.i+Jumlah.u+Jumlah.o+Jumlah.e;
n=n/Jumlah.charc;
n=n*100;
```

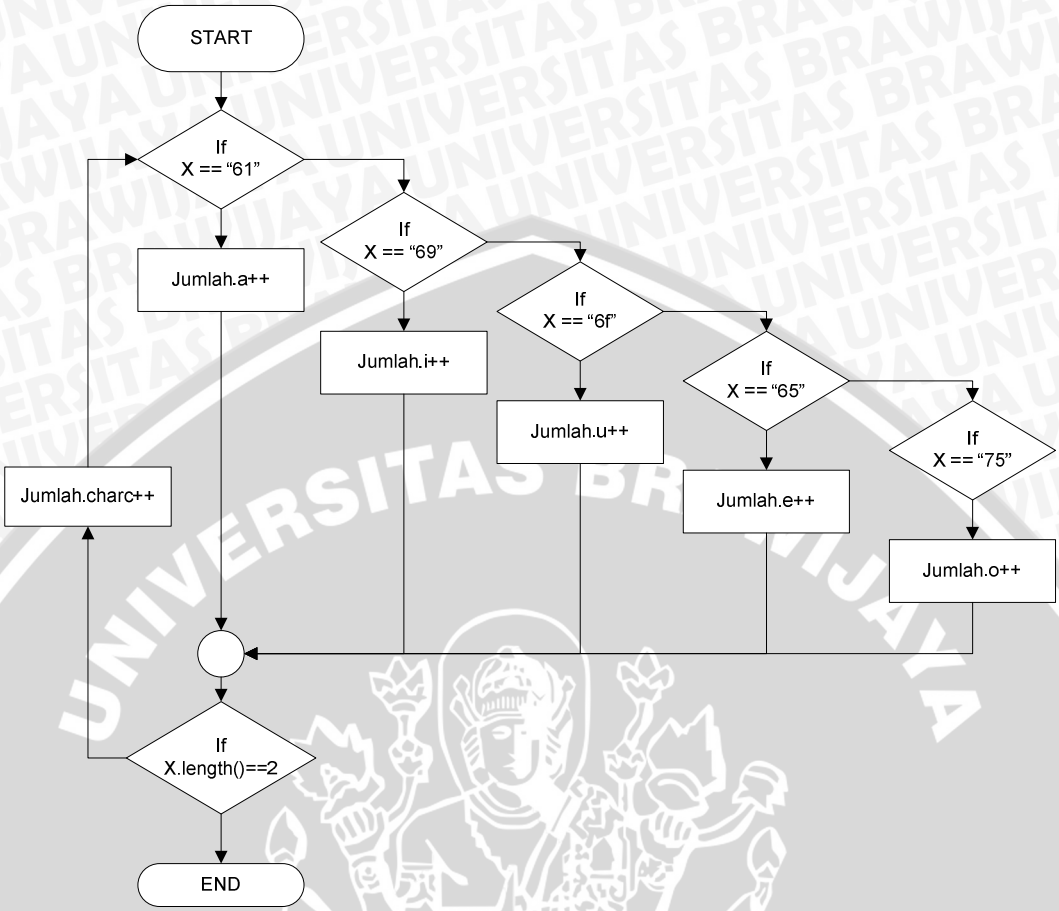
```
        cout<<">> ANALISIS JENIS PAKET ADALAH :"<<n<<" %
Teks<<<"<<endl;
return 0;
}

void reset ()
{
        Jumlah.a=0;
        Jumlah.i=0;
        Jumlah.u=0;
        Jumlah.e=0;
        Jumlah.o=0;
        Jumlah.charc=0;

}
void checkHuruf(string x){
        if (x=="61")
                Jumlah.a++;
        else if (x=="69")
                Jumlah.i++;
        else if (x=="6f")
                Jumlah.u++;
        else if (x=="65")
                Jumlah.e++;
        else if (x=="75")
                Jumlah.o++;
        if (x.length()==2){
                Jumlah.charc++;
        }
        //cout<<x<<endl;
}
```


start
Deklarasi Variabel
Fungsi reset ()
Fstream.fbin;
Fbin.open(file, modus operasi IO)
!fbin
yes
no
Error message
“ coud not open file “
A



Gambar 5.4 *Flowchart* program analisis

Sumber : perancangan



Gambar 5.5 Algoritma fungsi checkhuruf

Sumber : perancangan

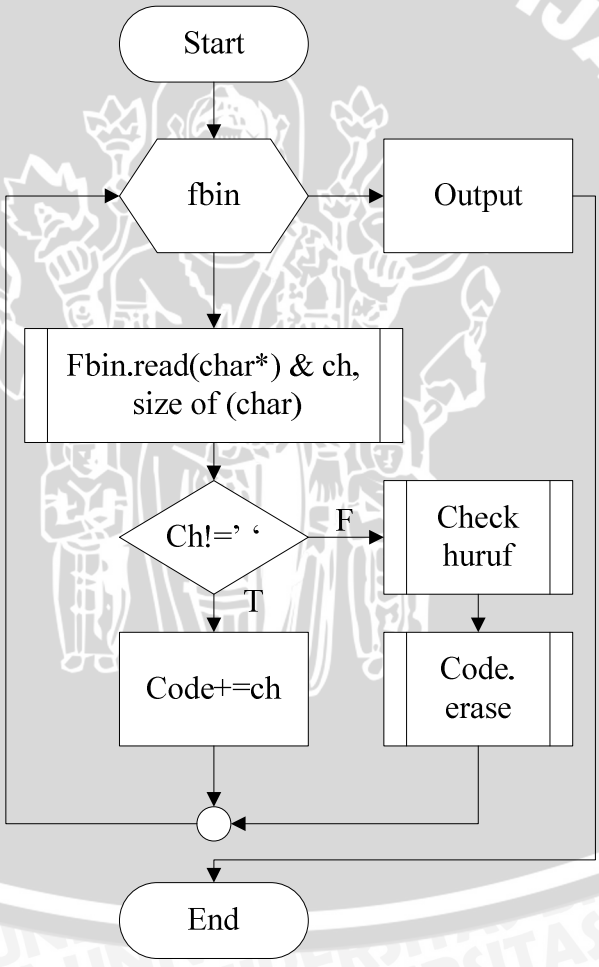
5.3 Implementasi Algoritma *Parsing* Data *File*.

5.3.1 Penentuan algoritma *parsing file*

Dalam penlitian kali ini, peneliti menggunakan metode penguraian atau *parsing* data untuk menganalisis data hasil perekaman. Data yang akan diurai adalah data *file ferdi.pcap.txt* yang merupakan keluaran dari program *tshark*. Sedangkan teknik yang digunakan dalam metode *parsing* itu sendiri adalah metode *top-down*.

5.3.2 Implementasi algoritma metode *parsing*

Implementasi metode algoritma *parsing file* ini dapat dijabarkan melalui diagram alir dibawah ini.



Gambar 5.6 Diagram alir metode *parsing file*.

Sumber: perancangan

Secara singkat, penjelasan dari diagram alir algorima penguraian data diatas adalah sebagai berikut :

1. *Parsing* menggunakan parameter spasi – karakter. Setiap pemindaian data telah menemukan spasi, karakter sebelum spasi yang dikumpulkan dimasukan kedalam fungsi *checkhuruf*, setelah it maka *buffer string*-nya akan dihapus.
2. Pada fungsi *checkhuruf*, kumpulan karakter tersebut akan dicocokan dengan karaker yang mewakili huruf a, i, u, e, o dan apabila ditemukan kecocokan maka *counter* huruf tersebut akan ditambah.
3. Kemudian *string* (kumpulan karakter) yang telah ditambahkan tadi akan diperiksa panjang *string*-nya (banyaknya karakter dalam *string*). Apabila panjangnya 2 karakter, maka akan dianggap karakter *body*, lalu menambah *counter* jumlah karakter. Apabila panjangnya tidak sama dengan 2 karakter, maka akan dianggap bagian selain *body*.

5.4 Implementasi Pengambilan Asumsi Program

Pada bagian ini, peneliti akan mengambil asumsi untuk keluaran program. Asumsi ini bertujuan agar keluaran program dapat mengarah ke hasil yang diharapkan. Pada prosesnya, peneliti menggunakan langkah-langkah sebagai berikut :

1. Paket data yang dilewatkan berasal dari server FTP *(File Transfer Protokol*).
2. Sumber data adalah *file* percobaan.txt
3. Topologi jaringan yang digunakan sesuai dengan perancangan yang dijelaskan sebelumnya.
4. Paket yang ditangkap sebanyak 4 paket.
5. Percobaan akan dilakukan sebanyak tiga kali agar data yang didapat lebih baku.

Percobaan	Paket data	Nilai Keluaran (%)	M (%)	m (%)
1	1	12,02984		
	2	8,75835		
	3	17,83654		
	4	26,73846	26,73846	8,75835
2	1	28,13578		
	2	24,95463		
	3	11,98474		
	4	32,00234	32,00234	11,98474
3	1	34,74663		
	2	12,44750		
	3	16,45637		
	4	29,35490	34,74663	12,44750
∑		255,4461	93,48743	33,19059

Tabel 5.1 Tabel hasil percobaan

Sumber : perancangan

Tahap berkutnya adalah menghitung nilai rata-rata maksimal, minimal, dan nilai rata-rata keluaran program.

- $\bar{M}$ = 93,48743 / 3 = 31,16247 %

- $\bar{m}$ = 33,19059 / 3 = 11,06353 %

- $\bar{n}$ = 255,4461 / 12 = 21,28717 %

- Range nilai hasil keluaran program untuk paket data teks (x) adalah :

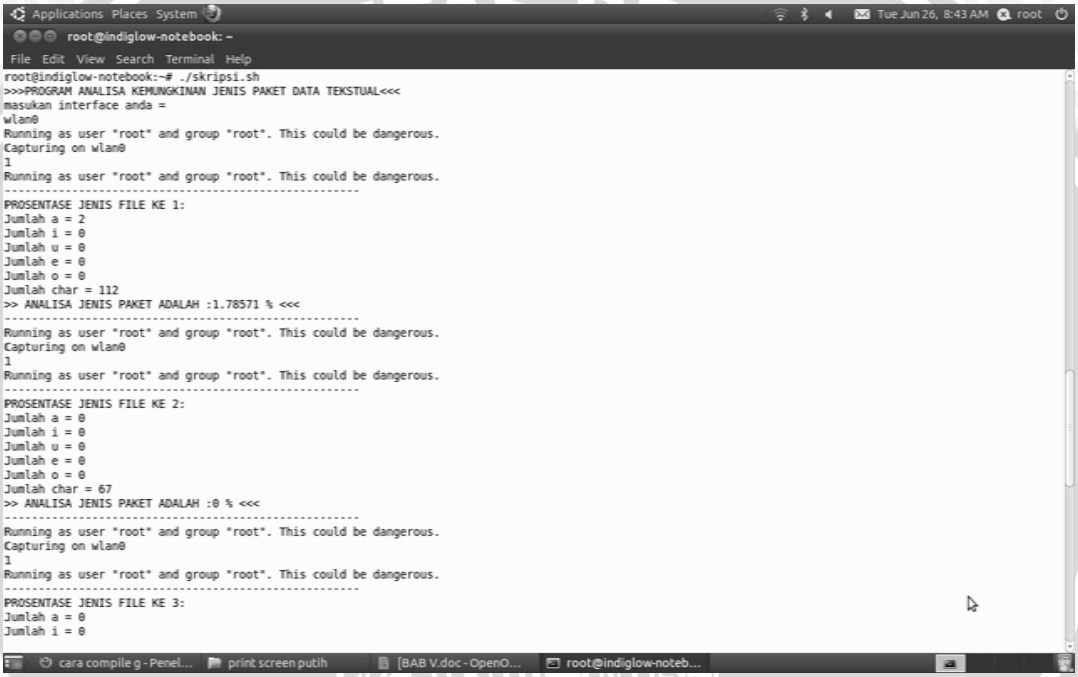
11,06353 % < x < 31,16247 %

Jadi, dari data percobaan diatas dapat ditarik kesimpulan bahwa jangkauan dari data asumsi adalah pada angka 11,06353 % - 31,16247 % dengan nilai rata-rata sebesar 21,28717 % . Artinya bahwa jika hasil analisis dari program untuk suatu paket data berada pada di dalam skala nilai diatas, maka paket data tersebut dianggap berjenis teks, dan jika berada di luar skala nilai tersebut maka paket tersebut dianggap non teks.

5.5 Implementasi Algoritma Program Gabungan (*skripsi.sh*)

Setelah menyelesaikan proses pembacaan paket data dan analisis perhitungannya, maka diperlukan sebuah konsep gabungan akhir yang akan digunakan untuk menggabungkan penjadwalan kerja secara otomatis. Jadi dalam pembahasan kali ini akan diterapkan teknik *bash scripting* yang akan memerintahkan sistem operasi untuk melakukan perintah seperti *console terminal, sequential*, dan berulang sesuai konsep. *Output* dari *bash script* ini adalah file *skripsi.sh* yang dapat dieksekusi dengan perintah :

```
root@indiglow# ./skripsi.sh.
```

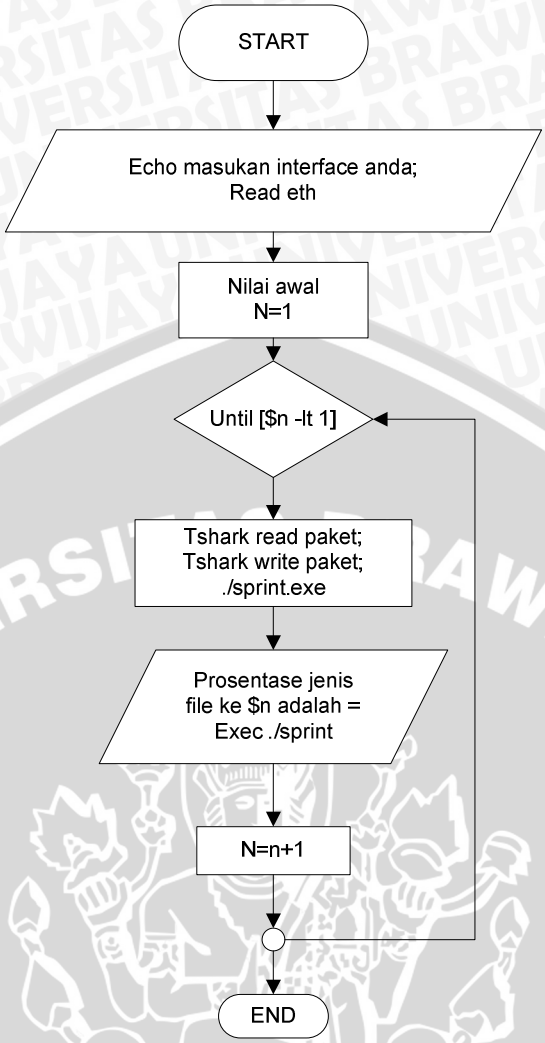


Gambar 5.7 Gambar tampilan program *skripsi.sh*

Sumber : Perancangan

Syntax program skripsi.sh

```
#!/bin/bash
echo ">>>PROGRAM ANALISIS KEMUNGKINAN JENIS PAKET DATA TEKS<<<"
echo masukan interface anda =
read eth
n=1
until [ $n -lt 1 ];
do
tshark -c 1 -i $eth -w ferdi.pcap
tshark -x -r ferdi.pcap > ferdi.pcap.txt
program=./sprint
echo "-------------------------------------------------"
echo  "PERSENTASE JENIS FILE KE $n:"
hasil=hsl | exec ./sprint
echo "-------------------------------------------------"
let n=n+1
done
```



Gambar 5.8 *Flowchart program skripsi.sh*

Sumber : Perancangan

BAB VI

PENGUJIAN SISTEM

Bab ini membahas mengenai pengujian dan analisis terhadap implementasi Program Analisis Kemungkinan Jenis Paket Data Teks. Pengujian yang dilakukan meliputi pengujian topologi dan pengujian implementasi aplikasi sistem.

6.1. Pengujian Diagram Jaringan

Pengujian diagram jaringan ini bertujuan untuk mengetahui apakah diagram jaringan yang telah dibuat dapat terkoneksi satu sama lain. Pengujian ini dilakukan dari sisi *server*.

6.1.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah semua perangkat pada diagram jaringan yang telah dibuat telah dapat saling berkomunikasi satu sama lain.

6.1.2 Spesifikasi dan konfigurasi komputer

Komputer *server* memiliki spesifikasi sebagaimana yang telah disebutkan di bagian awal.

6.1.3 *Software* aplikasi

Software aplikasi untuk menguji kerja jaringan pada pembahasan kali ini akan menggunakan *ping*, dengan alamat tujuan yaitu salah satu *server web* luar atau komputer lain yang terhubung dalam jaringan.

6.1.4 Prosedur pengujian

Pengujian jaringan dilakukan dengan melakukan beberapa prosedur yaitu :

1. Memastikan komputer telah terhubung dengan internet.
2. Melakukan *ping* ke salah satu *web server* internet, misal : www.google.com .
3. Melakukan *ping* ke *gateway* komputer yang terhubung dengan jaringan lokal, misal : 99.99.99.1/24

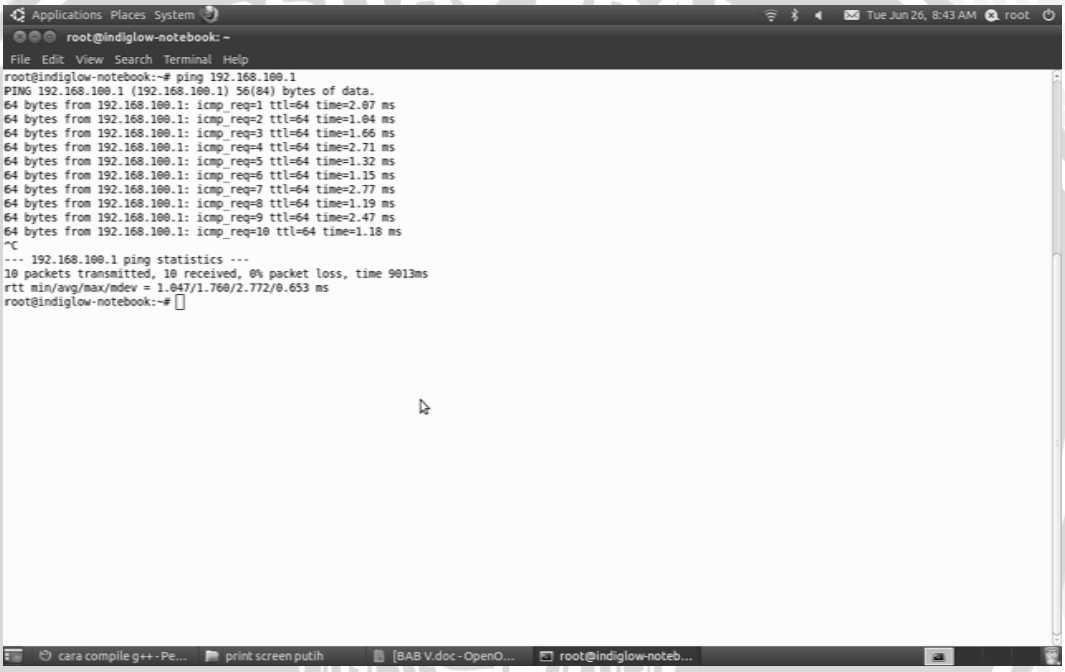
4. Memantau hasil *ping*, apakah *reply* dengan baik atau belum terhubung.

6.1.5 Hasil yang diharapkan

Hasil yang diharapkan dari pengujian ini adalah komputer *Server* mendapat balasan atas pesan yang dikirimkan ke alamart tujuan Selain itu komputer *server* dan jaringan luar dapat saling berkomunikasi.

6.1.6 Hasil pengujian dan analisis

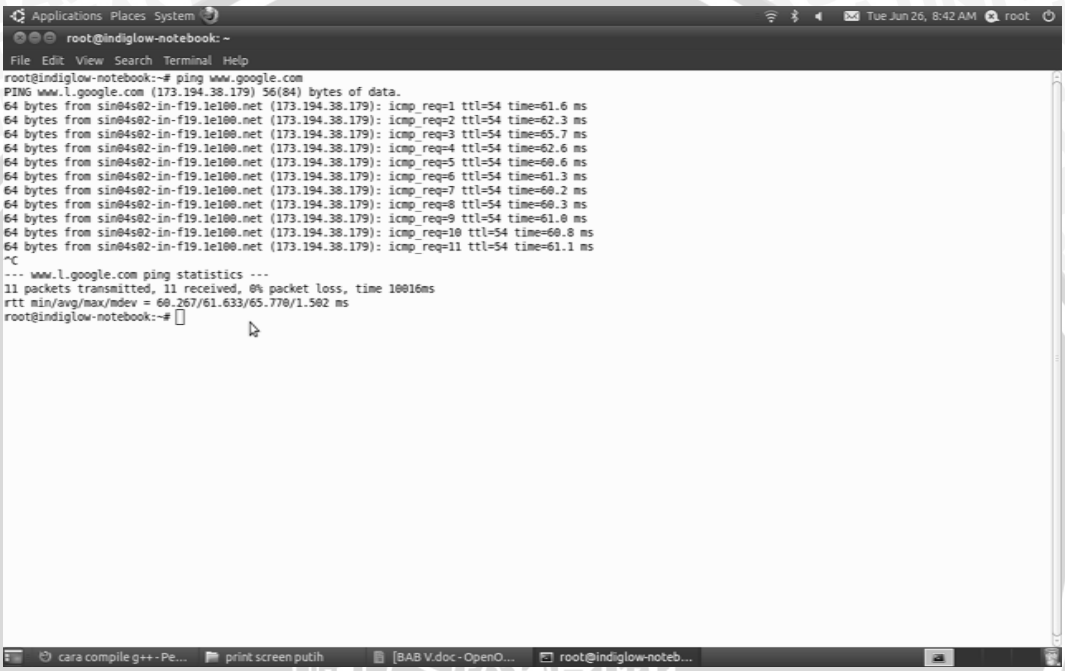
Saat perintah *ping* dijalankan, komputer *server* akan mengirim pesan *ping* ke seluruh alamat *IP network* yang bersangkutan. Alamat *IP* pada perintah *ping* akan mengirim pesan balasan. Hasil dari *ping* adalah sebagai berikut.



Gambar 6.1 Hasil Test *ping* ke *gateway*

Sumber : pengujian

Dari gambar diatas, dapat diketahui bahwa komputer *server* telah terhubung dengan jaringan luar dan lokal. Dapat dilihat bahwa hasil *ping* ke alamat *domain* www.google.com telah mendapatkan balasan. Begitu juga dengan hasil *ping* ke alamat 99.99.99.1/24 telah mendapatkan balasan dari alamat tujuan.



Gambar 6.2 Hasil Test *ping* ke alamat *web www.google.com*

Sumber : pengujian

6.1.7 Kesimpulan

Semua komputer *server* dan jaringan lokal maupun internet telah dapat berkomunikasi satu sama lain.

6.2. Pengujian Implementasi Program Analisis Kemungkinan Jenis Paket Data Teks

Pengujian ini bertujuan untuk mengetahui apakah program yang telah dirancang sebelumnya dapat bekerja sesuai yang diharapkan. Pengujian ini dilakukan dengan menggunakan sistem operasi *Linux Ubuntu Dekstop* dan bahasa pemrograman C yang ditempatkan pada komputer *server*. Pengujian implementasi disusun dengan cara komputer *server* berada dalam sebuah jaringan komputer.

6.2.1. Pengujian program perekam paket data

6.2.1.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah program perekeman paket data yang digunakan dalam hal ini adalah *tshark* telah bekerja dengan baik.

6.2.1.2 Prosedur pengujian

Pengujian dilakukan dengan melakukan beberapa prosedur yaitu :

1. Menjalankan program *tshark* dengan memberikan parameter –c, -w, -i; untuk memulai merekam paket data.

```
root@indiglow# tshark –i wlan0 –c 1 –w ferdi.pcap
```

2. Memeriksa apakah keluaran dari program *tshark* berupa *file ferdi.pcap* telah ada pada *folder ./root* .

3. Menjalankan program *tshark* untuk menulis data hasil perekaman ke dalam *file ferdi.pcap.txt*.

```
root@indiglow# tshark –x –r ferdi.pcap > ferdi.pcap.txt
```

4. Memeriksa apakah keluaran dari program *tshark* berupa *file ferdi.pcap.txt* telah ada pada *folder ./root* .

6.2.1.3 Hasil yang diharapkan

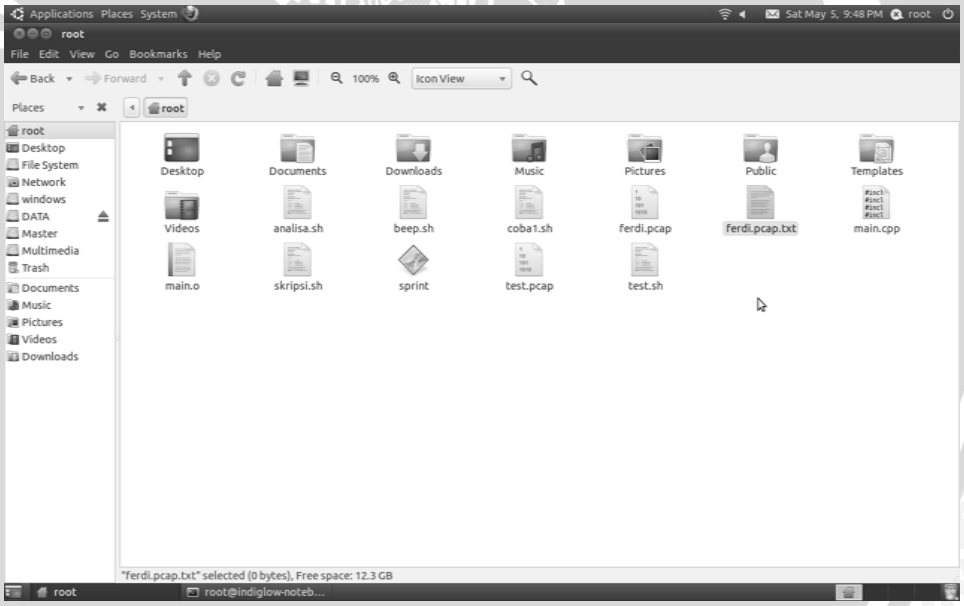
Hasil yang diharapkan dari pengujian ini adalah program *tshark* telah dapat bekerja dengan baik untuk merekam dan menulis paket data ke dalam file *ferdi.pcap.txt* untuk kemudian diolah.

6.2.1.4 Hasil pengujian dan analisis

Hasil dari pengujian diatas adalah bahwa program *tshark* telah dapat bekerja untuk merekam dan menulis isi paket data ke dalam sebuah *file ferdi.pcap.txt*.

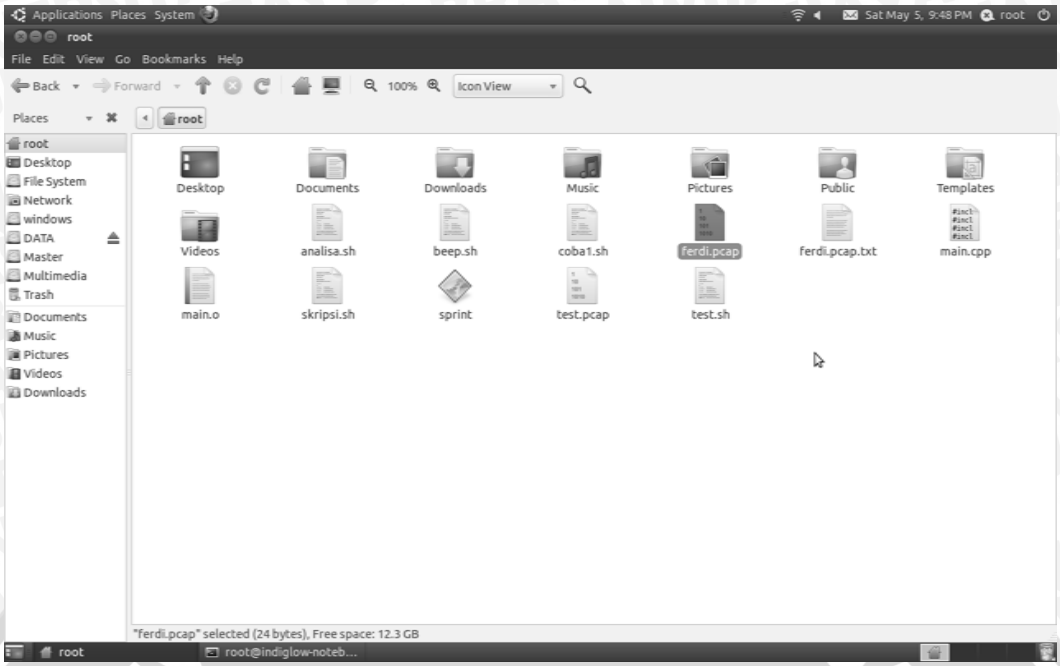
6.2.1.5 Kesimpulan

Dari hasil pengujian yang telah dilakukan di atas dapat disimpulkan bahwa Program *tshark* telah bekerja untuk merekam dan menulis isi paket data ke sebuah file *ferdi.pcap.txt*. Hal ini dapat diperiksa untuk ketersediaan *file* yang dimaksud pada *folder ./root* .



Gambar 6.3 *File ferdi.pcap.txt* hasil pembacaan *file ferdi.pcap*.

Sumber : pengujian



Gambar 6.4 *File ferdi.pcap* hasil rekaman paket data

Sumber : pengujian

6.2.2. Pengujian program *sprint.exe*

Pengujian implementasi perangkat lunak *sprint.exe* adalah untuk mengetahui apakah program dibangun sudah benar sesuai dengan yang dibutuhkan. Kebutuhan-kebutuhan yang telah dirumuskan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian implementasi.

6.2.2.1 Tujuan

Sistem dapat menunjukkan bahwa program analisis jenis paket data teks ini dapat menghitung persentase kemungkinan jenis paket data teks.

6.2.2.2 Prosedur pengujian

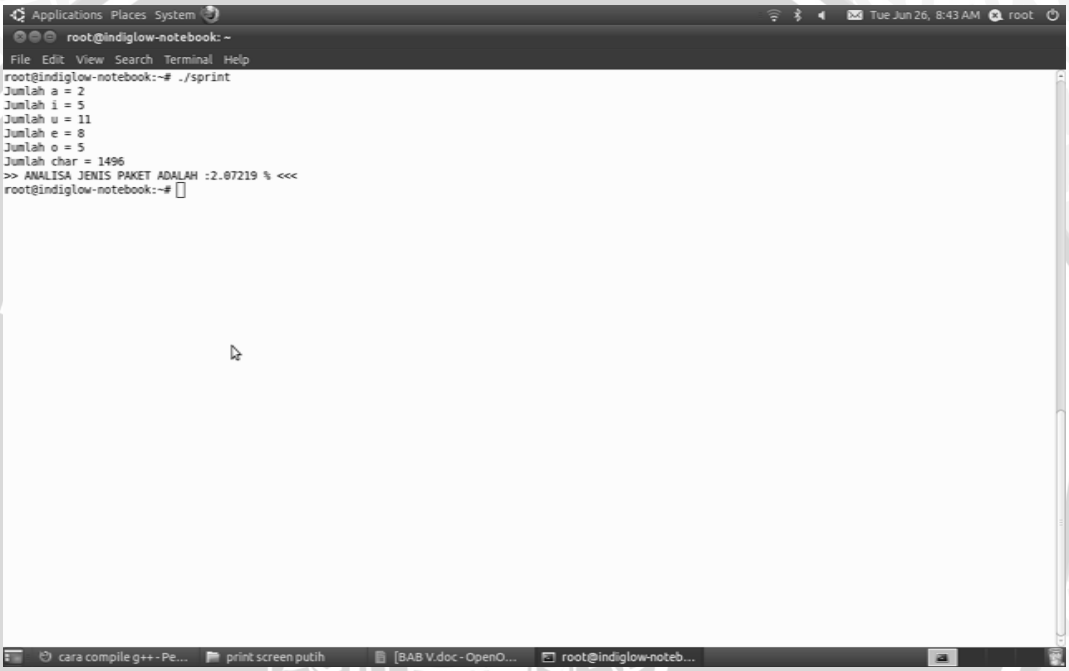
1. Melakukan pengujian dengan memberikan masukan dari program perekam data.
2. Melihat apakah program dapat menampilkan hasil perhitungan dengan baik.

6.2.2.3 Hasil yang diharapkan

Hasil yang diharapkan dari pengujian ini adalah mendapatkan hasil tentang kinerja program analisis paket data, agar tujuan penelitian dapat tercapai.

6.2.2.4 Hasil pengujian dan analisis

Dari pengujian yang telah dilakukan dapat dihasilkan angka-angka yang menunjukan besarnya persentase kemungkinan munculnya jenis paket data tesk atau non teks. Berikut merupakan tampilan dari keluaran program.



Gambar 6.5 Tampilan keluaran program *Sprint.exe*

Sumber : pengujian

6.2.2.5 Kesimpulan

Dari hasil pengujian yang telah dilakukan di atas dapat disimpulkan bahwa program telah dapat menghitung besarnya persentase kemungkinan jenis paket data teks.

6.3 Pengujian Implementasi Program Gabungan

Program gabungan ini dinamakan *skripsi.sh* . Program ini berisi kumpulan perintah dan *module* program *sprint.exe*. Program ini mengatur jalannya mekanisme mulai dari pembacaan data hingga keluaran program.

6.3.1 Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui apakah program telah bekerja sesuai yang diharapakan. Program ini diharapkan mampu memberikan keluaran sesuai rancangan.

6.3.2 Prosedur pengujian

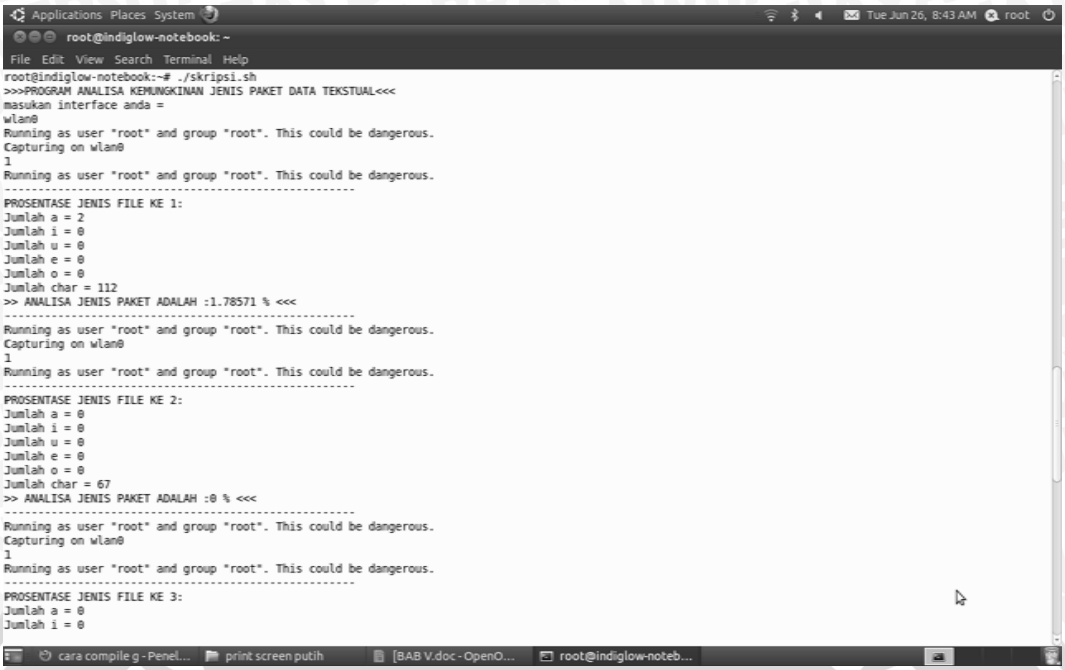
1. Pengujian dilakukan dengan menjalankan program *skripsi.sh*
2. Memberikan masukan program berupa keterangan *ethernet* yang akan digunakan.
3. Melihat hasil keluaran program.

6.3.3 Hasil yang diharapkan

Dari pengujian ini diharapkan program dapat bekerja memadukan proses pembacaan paket data, penulisan paket data, hingga kalkulasi data oleh *sprint*.exe untuk kemudian menampilkan hasil keluaran yang berupa angka persentase.

6.3.4 Hasil pengujian dan analisis

Dari pengujian ang telah dilakukan, didapatkan hasil bahwa program dapat bekerja untuk memadukan proses pembacaan data, penulisan data, dan menjalankan program *sprint*.exe untuk kalkulasi data hasil rekaman.



Gambar 6.6 Tampilan hasil keluaran program *skripsi.sh*

Sumber : pengujian

6.3.5 Kesimpulan

Dari pengujian diatas dapat ditarik kesimpulan bahwa program telah bekerja dengan baik sesuai hasil yang diharapkan, yaitu memadukan proses pembacaan paket data, penulisan paket data, hingga kalkulasi data oleh *sprint.exe* untuk kemudian menampilkan hasil keluaran yang berupa angka persentase.

6.4 Pengujian Performansi Program Terhadap Jaringan

6.4.1 Tujuan

Tujuan dari pengujian performansi program terhadap jaringan ini adalah untuk mengetahui sejauh mana kemampuan program terhadap parameter jaringan yaitu lebar jalur atau *bandwidth*.

6.4.2 Prosedur pengujian

Untuk melakukan pengujian kali ini, penulis menggunakan prosedur sebagai berikut.

1. Pengujian dilakukan dengan mengakses data *html* dari *web-server* pada jaringan lokal dari server analisis.
2. Program yang akan diuji adalah *skripsi.sh*
3. Parameter untuk jumlah total paket yang dilewatkan adalah sebanyak 100 paket data untuk tiap skala nilai.
4. Skala *bandwidth* yang digunakan adalah 10 Kbps hingga 1000 Kbps.
5. Protokol komunikasi yang digunakan adalah *TCP/IP* dengan format data *HTML*.
6. Hasil perhitungan skala nilai adalah dalam satuan persen (%).
7. Perhitungan hasil uji perfomansi mengikuti rumus :
   SPP = (Nilai tangkap : Parameter paket) x 100%
   SPP : Skala Persentase Performa.
   Nilai Tangkap : Nilai hasil pengukuran.
   Parameter paket : Jumlah paket data yang digunakan dalam pengukuran sebesar 100 paket data.

6.4.3 Hasil yang diharapkan

Dari pengujian yang dilakukan, diharapkan peneliti mendapatkan skala nilai hasil pengukuran yang menunjukan nilai perfomansi program, hingga diketahui titik puncak perfomansi dan bentuk grafiknya.

6.4.4 Hasil pengujian dan analisis

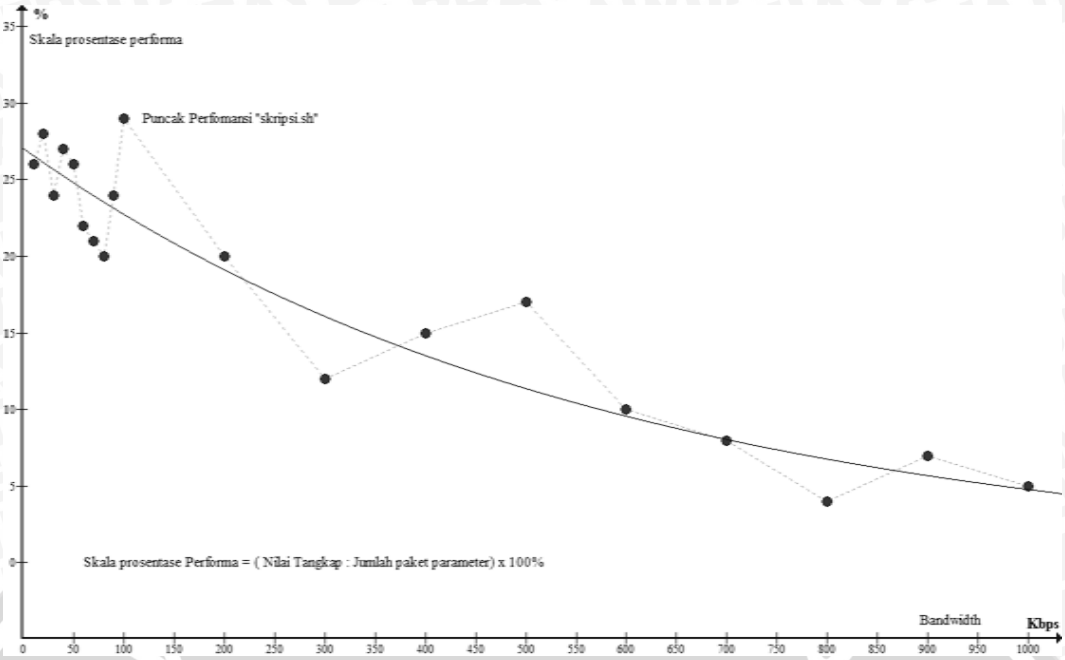
Dari hasil pengujian, didapatkan hasil sebagai berikut :

Bandwidth (Kbps)	**Hasil Pengukuran (%)**	**Bandwidth (Kbps)**	**Hasil Pengukuran (%)**
10	26	**200**	20
20	28	**300**	12
30	24	**400**	15
40	27	**500**	17
50	26	**600**	10
60	22	**700**	8
70	21	**800**	4
80	20	**900**	7
90	24	**1000**	5
100	29		

Tabel 6.1 Tabel hasil pengujian program.

Sumber : pengujian

Dari hasil pengukuran di atas, didapatkan hasil bahwa nilai hasil pengukuran cenderung berada di kisaran 5 % hingga 29 % dan puncak performa program berada pada angka 100Kbps.



Grafik 6.1 Grafik performa program terhadap bandwidth untuk program *skripsi.sh*

Sumber : pengujian

6.4.5 Kesimpulan

Dari hasil pengujian diatas, dapat disimpulkan bahwa nilai persentase performa program berada pada kisaran nilai 5 % - 29 % , sedangkan puncak performa program berada pada nilai bandwidth 100 Kbps.

6.5 Pengujian Program *skipsi.sh* Terhadap Paket Data Acak

6.5.1 Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui hasil kerja program terhadap paket data acak apakah dapat memenuhi asumsi atau tidak. Paket data akan dikirimkan secara acak, artinya dari segi jenis sumber paket datanya tidak terpaku pada jenis file teks.

6.5.2 Prosedur pengujian

Untuk melakukan pengujian ini, maka diperlukan prosedur untuk mengatur jalannya pengujian agar mendapatkan hasil seperti yang diharapkan.

1. Merancang topologi jaringan sesuai perancangan yang telah dijelaskan pada bab sebelumnya.
2. Melakukan aktifitas *browsing* data menggunakan *browser*, seperti *Mozilla firefox, Internet Explorer*, dll.
3. Mencatat hasil keluaran program sebanyak 30 paket data.
4. Menganalisis hasil keluaran program berdasarkan asumsi :

   $$11{,}06353\ \% < x < 31{,}16247\ \%$$

5. Jika nilai x berada pada jangkauan asumsi pada poin 4, maka x dianggap berjenis teks, dan jika tidak maka x dianggap non teks.

6.5.3 Hasil yang diharapkan

Dari pengujian yang dilakukan, diharapkan peneliti mendapatkan skala nilai hasil pengukuran dari sumber paket data acak. Sehingga dapat diketahui jenis kemngkinan paket data tersebut.

6.5.4 Hasil pengujian dan analisis

Dari pengujian yang dilakukan diatas, didapatkan hasil seperti yang tercantum dalam tabel dibawah ini.

Paket paket data ke x	Hasil Keluaran %	Kemungkinan Jenis Paket Data
1	0,64725	Non teks
2	0,75848	Non teks
3	0,12785	Non teks
4	4,78473	Non teks
5	8,99983	Non teks
6	34,75624	Teks
7	1,22376	Non teks
8	12,75632	Teks
9	8,94637	Non teks
10	17,48372	Teks
11	21,87463	Teks
12	22,12876	Teks
13	1,98743	Non teks
14	0,54637	Non teks
15	4,65473	Non teks
16	6,98746	Non teks
17	0,64532	Non teks
18	0,44567	Non teks
19	1,56372	Non teks
20	2,38976	Non teks
21	1,88756	Non teks

22	7,34672	Non teks
23	16,98364	Teks
24	24,28752	Teks
25	12,17465	Teks
26	0,19384	Non teks
27	4,57822	Non teks
28	7,98365	Non teks
29	28,55467	Teks
30	24,76573	Teks

Tabel 6.2 Data hasil pengujian dari paket data acak.

Sumber : pengujian

Dari data percobaan diatas dapat diketahui bahwa program telah menganalisis 30 paket data. Dari 30 paket data yang dianalisis, 10 diantaranya adalah diperkirakan berjenis teks, dan sisanya sebanyak 20 paket data berjenis non teks. Hal ini sesuai dengan asumsi yang diberikan pada pembahasan bab sebelumnya.

6.5.5 Kesimpulan

Kesimpulan dari data pengujian diatas adalah bahwa program telah bekerja dengan baik untuk menganilisa jenis paket data berdasarkan asumsi yang diberikan.

BAB VII

PENUTUP

7.1. Kesimpulan

Dari hasil pengujian dan analisis Program Analisis Kemungkinan Jenis Paket Data Teks didapatkan kesimpulan :

1. Aplikasi program untuk analisis isi paket data telah berhasil dibuat, untuk menghitung kemungkinan sebuah sebuah paket data apakah berjenis teks atau non-teks. Besarnya nilai kemungkinan ditunjukan dalam skala persentase.
2. Pengujian program analisis jenis paket data teks telah dilaksanakan dengan baik. Dalam pengujian dapat ditarik kesimpulan bahwa hasil yang ditunjukan tidak dapat sepenuhnya tepat karena hanya merupakan nilai pendekatan.
3. Besar nilai hasil pengujian dan analisis data oleh program dapat diperoleh bahwa nilai hasil pengukuran berkisar antara 5% - 29%.
4. Puncak performa program terhadap jaringan pada format data html adalah pada skala bandwidth 100 Kbps sebesar 29 % dengan bentuk grafik performa program eksponensial.
5. Hasil pengukuran untuk keluaran program (x) didapatkan *range* sebesar 11,06353 % < x < 31,16247 % untuk setiap paket data.

7.2. Saran

Saran yang dapat diberikan untuk pengembangan Program Analisis Jenis Paket Data Teks ini adalah:

1. Mengembangkan program analisis jenis paket data teks ini ke arah filter paket data ataupun firewall.
2. Guna meningkatkan dan memaksimalkan kinerja dari program, maka bisa dikembangkan algoritma atau mekanisme kerja program yang lebih baik.

DAFTAR PUSTAKA

[STEV-94] Stevens W. Richard, TCP/IP Illustrated Volume 1 The Protocols, Addison-Wesley Professional Computing Series, 1994

[GSB-11]Andri Kurniawan, 2009. http://gudang-sejarah.blogspot.com/2009/11/sejarah-bahasa-pemograman-c.html. (Diakses tanggal 12 Juli 2012).

[WPD-07] Wikipedia, 2007. ASCII. http://id.wikipedia.org/wiki/ASCII.(diakses tanggal 4 November 2011).

[WPD-12] http://id.wikipedia.org/wiki/transmission_control_protocol. (diakses tanggal 4 November 2011).

[WPD-04] http://id.wikipedia.org/wiki/Jaringan_komputer.(Diakses tanggal 5 November 2011)

[USU-11] http://usupress.usu.ac.id/files/Pemrograman Komputer; Satu Pendekatan kepada Pemrograman Berorientasikan Objek dalam C++ - Final_bab 1.pdf. (Diakses tanggal 4 November 2011).

[ALG-11] http://id.wikipedia.org/wiki/Algoritma. (Diakses tanggal 6 November 2011).

[EPR-90] Nurhayati, Dwi Okky.Dasar Algoritma. http://eprints.undip.ac.id/18630/1/pertemuan2. pdf. (Diakses tanggal 6 November 2011).

[MOH-11] Mohiqbal.2010.Linux dan Keamanan. http://mohiqbal.staff.gunadarma.ac.id. (Diakses tanggal 8 November 2011).

[WPD-05] wikipedia.2011.Binary file.http://en.wikipedia.org/wiki/Binary_file (diakses tanggal 8 November 2011)

[PRS-09] it.gunadarma. 2010. http ://www.ti-menengah.lab.gunadarma.ac.id/wp.../BAB-VI-Metode-Parsing.pdf (Diakses tanggal 12 Juli 2012)