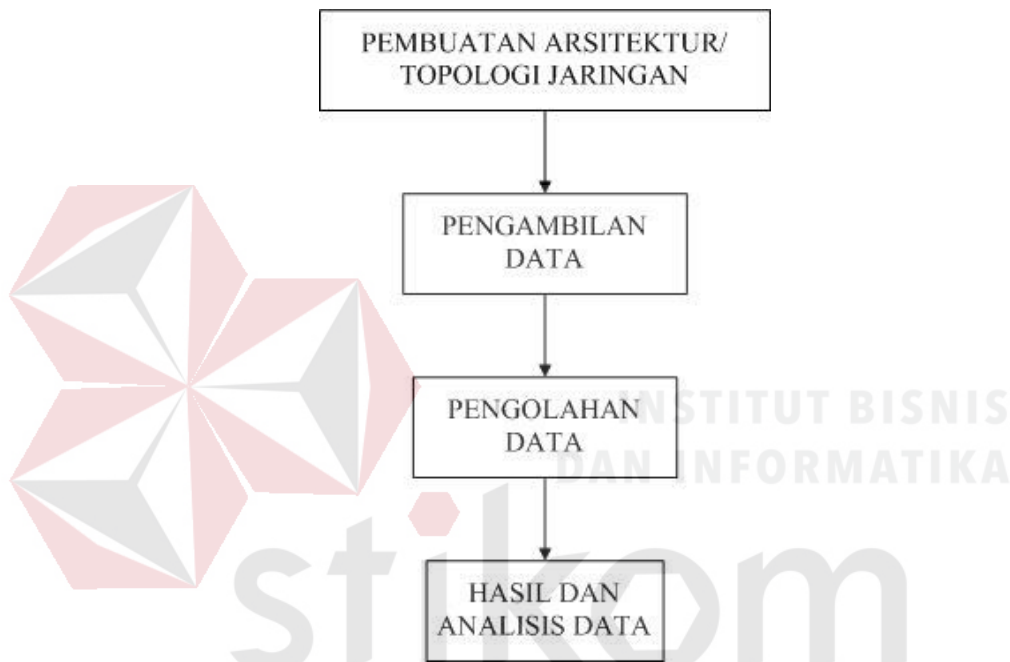


BAB III

METODOLOGI PENELITIAN

3.1. Perancangan Sistem dan Blok Diagram Sistem

Perancangan sistem dan diagram dapat dijelaskan dengan baik seperti yang terlihat di gambar 3.1.

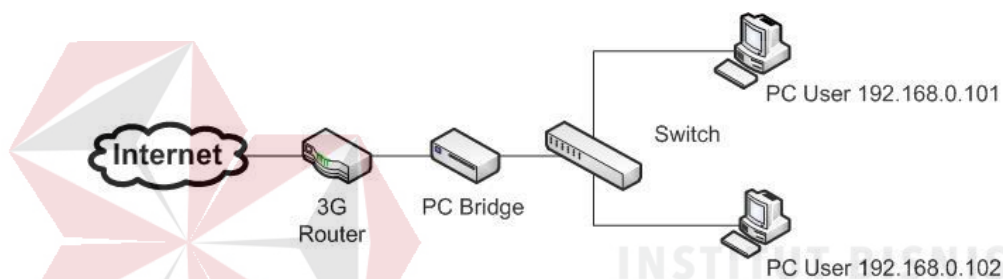


Gambar 3.1. Blok Diagram Sistem

Dapat kita ketahui cara kerja dan metodologi penelitian yang dilakukan dari blok diagram sistem pada gambar 3.1. Mulai dari pembuatan hingga hasil analisis yang telah dihitung nilai-nilai parameter *Quality of Service* (QoS) yaitu Utilisasi *Bandwidth*, *Delay*, *Jitter* dan *Packet Loss*. Metode penelitian yang digunakan adalah studi kepustakaan dan analisis karakteristik lalu lintas data internet pada aplikasi *web social network* serta hasil QoS-nya yang mencakup *facebook* dan *twitter*. Oleh karena itu, dikumpulkan data dan informasi serta materi dasar yang bersifat teoritis yang sesuai dengan permasalahan.

3.2. Arsitektur Jaringan (Topologi)

Dalam sebuah perancangan sistem dibutuhkan komponen untuk pembuatan arsitektur atau topologi jaringan. Komponen yang dimaksud yaitu perangkat keras. Arsitektur jaringan di desain sedemikian rupa agar dapat dilakukan penelitian. Dengan adanya perancangan desain arsitektur jaringan ini maka akan memudahkan dalam penyusunan komponen arsitektur jaringan, sekaligus memudahkan untuk pengambilan data yang akan digunakan untuk penelitian. Berikut gambar arsitektur jaringan dapat dilihat di gambar 3.2.



Gambar 3.2. Arsitektur Jaringan

Komponen dalam arsitektur jaringan terdiri dari :

1. 1 buah modem Huawei E160 dengan kecepatan *maximum* 7,2 Mbps, dengan menggunakan kartu 3 (*three*) sebagai layanan *provider internet*, yang terpasang di sebuah *router*.
2. 1 buah *router* TP-Link MR 3420.
3. 1 buah *Switch* TP-Link 5 Port
4. 1 buah PC komputer yang digunakan sebagai *Bridge*.
5. 2 buah *PC User*, masing-masing memiliki IP 192.168.0.101 untuk *user* pertama dan IP 192.168.0.102 untuk *user* yang kedua seperti terlihat pada gambar 3.2.

Untuk terhubung ke internet, arsitektur jaringan yang dibuat yaitu menggunakan *router* yang telah dipasang sebuah modem dengan layanan *provider*

kartu 3 (*three*) sebagai jasa layanan *internet* dengan kecepatan hingga 3,6 Mbps untuk sinyal 3G/HSDPA dan kecepatan hingga 384 Kbps untuk sinyal EDGE. Alasan menggunakan layanan internet kartu 3 (*three*) ini karena *provider* ini cukup baik sinyalnya dan memiliki kecepatan serta kapasitas kuota yang lumayan besar.

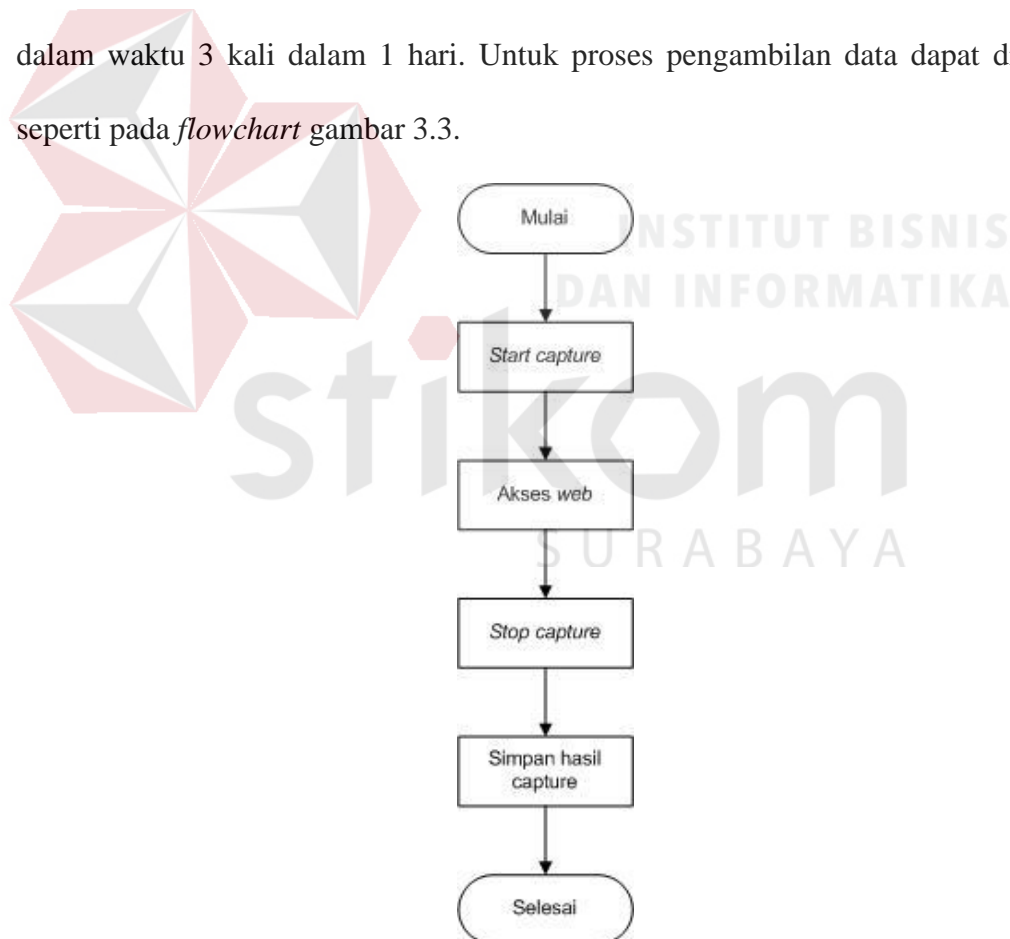
Pada arsitektur jaringan yang dibuat, modem yang dipasang ke sebuah *router*. *Router* tersebut kemudian dihubungkan dengan *PC Bridge*. *PC Bridge* yaitu sebuah komputer/PC yang mempunyai 2 buah *interface* NIC (*Network Interface Card*). Satu *interface* pada sebuah komputer *bridge* dihubungkan ke *router*, sedangkan *interface* yang lain dihubungkan ke sebuah *switch* untuk menghubungkan dengan *PC user* yang ada. *PC bridge* ini digunakan untuk tujuan melihat lalu lintas data/ paket data yang melintas pada saat *user* mengakses semua aplikasi *web* yang berada pada internet. Lalu lintas data tersebut antara *client* dan *server*. Dengan menggunakan aplikasi *Network Protocol Analyzer* yaitu *Wireshark* yang dipasang pada komputer/PC *Bridge*, maka dapat dilakukan penangkapan lalu lintas paket data yang lewat. *Wireshark* digunakan karena merupakan aplikasi *Network Protocol Analyzer* yang dapat digunakan pada *Operating System* (OS) *Windows*, karena *PC bridge* yang digunakan menggunakan *Operating System* (OS) *Windows*.

Pada *PC user* 192.168.0.101 dan *PC user* 192.168.0.102 yang terhubung ke *switch* digunakan untuk mengakses beberapa aplikasi *web* secara bersama-sama dalam 1 waktu. Beberapa aplikasi *web* yang diakses diantaranya *facebook*, *twitter*, *youtube* dan sesekali membuka *web-web* yang lain.

Untuk arsitektur jaringan yang digunakan bersifat tetap tidak berpindah-pindah tempat, hal ini dikarenakan *network coverage* dari sinyal HSDPA milik *provider* internet kartu 3 (*three*) terbatas. Tempat pengambilan data dilakukan ditempat tinggal penulis, yang ter-*coverage* sinyal HSDPA kartu 3 (*three*).

3.3. Pengambilan Data

Pada pengambilan data, metode yang digunakan adalah sampling data pada saat *user* mengakses aplikasi *web social network facebook* dan *twitter*. Sampling data tersebut didapat dari 2 buah *user* yang mengakses internet selama 1 minggu, dalam waktu 3 kali dalam 1 hari. Untuk proses pengambilan data dapat dilihat seperti pada *flowchart* gambar 3.3.



Gambar 3.3. *Flowchart* pengambilan data

Dari *flowchart* gambar 3.3 dapat dijelaskan dalam pengambilan data dengan menggunakan aplikasi *Network Protocol Analyzer* yaitu *wireshark* yang dipasang

pada komputer/PC bridge. Agar dapat dilakukan penangkapan lalu lintas paket data yang lewat, maka pada aplikasi *wireshark* pilih *interface bridge* pada saat memulai *start capture*.

Pada saat pengujian, pengambilan data yang di *capture* tidak hanya aplikasi *web social network* saja, tetapi juga dengan aplikasi *web video streaming*. Setelah selesai *stop capture*, data disimpan dan selanjutnya data mulai diolah kemudian dihitung untuk di analisis.

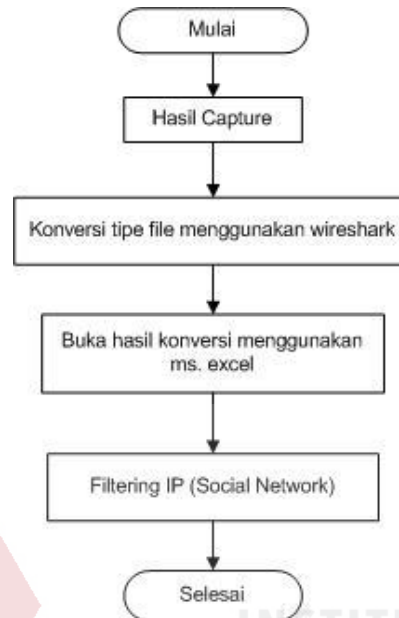
Pengambilan data dilakukan selama 1 minggu dalam waktu 3 kali dalam 1 hari yaitu pagi (08.00 – 10.15), siang (11.00 – 13.15) dan sore (15.00 – 17.15). Dimana waktu pagi orang-orang baru memulai jam kerja jadi lebih sedikit menggunakan aplikasi *web*, sedangkan waktu siang dimana orang-orang sedang istirahat sehingga menggunakan macam-macam aplikasi *web*, serta waktu sore dimana orang-orang pulang kerja sehingga lebih sedikit menggunakan aplikasi *web* yang ada dari waktu siang hari.

3.4. Pengolahan Data

Dari *flowchart* gambar 3.4 dapat dijelaskan dari data yang di *capture* menggunakan aplikasi *wireshark* disimpan dalam format file tertentu yang hanya bisa dibaca oleh aplikasi *wireshark* itu sendiri. Oleh karena itu, agar datanya bisa diakses aplikasi yang lain untuk diolah kemudian dihitung, file tersebut harus di-*export* ke bentuk file yang bisa dibaca oleh aplikasi lain, salah satu aplikasinya yaitu *Microsoft Excel*.

Setelah hasil *capture* dapat terbaca oleh *Microsoft Excel*, barulah dapat dilakukan pemrosesan data yang akan digunakan untuk analisis. Dalam analisis ini, diperlukan data yang berkaitan dengan aplikasi *web social network*, yang

mencakup *facebook* dan *twitter*. Untuk membedakan hasil yang telah di-*capture*, maka melakukan *filtering* tipe data yang di inginkan berdasarkan IP dari aplikasi *web social network facebook* dan *twitter*.



Gambar 3.4. *Flowchart* pengolahan data

Dalam penelitian 2 buah aplikasi *web social network* yaitu *facebook* dan *twitter*, maka IP yang di-*filter* hanya berdasarkan IP aplikasi *web social network*, yaitu IP *facebook* dan juga IP *twitter*. Cara mengetahui masing-masing IP aplikasi *web social network* yang akan digunakan untuk analisis adalah dengan memasukkan alamat *web* atau IP *web* dari aplikasi *web* yang didapat pada saat pengambilan data menggunakan *Wireshark*, ke aplikasi *web* pencari IP *address/ Domain Name Server Lookup (DNS Lookup)*, yang beralamatkan di <http://www.dnsstuff.com> atau <http://www.cqcounter.com/whois> atau <http://whois.domaintools.com>.

Berikut tabel 3.1 adalah daftar tabel IP aplikasi *web social network* yang telah didapatkan pada saat pengambilan data menggunakan *Wireshark* dan telah diketahui IP-nya melalui *DNS lookup* :

Tabel 3.1. IP address facebook dan twitter

Nama Web	IP adres (Net range)
www.facebook.com	<ul style="list-style-type: none"> ➤ 173.252.64.0 – 173.252.127.255 ➤ 31.13.64.0 – 31.13.127.255 ➤ 69.171.224.0 – 69.171.255.255
www.twitter.com	<ul style="list-style-type: none"> ➤ 199.59.148.0 – 199.59.151.255

3.5. Perhitungan Data

QoS adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwidth*, mengatasi *jitter* dan *delay* (Iversen, 2010).

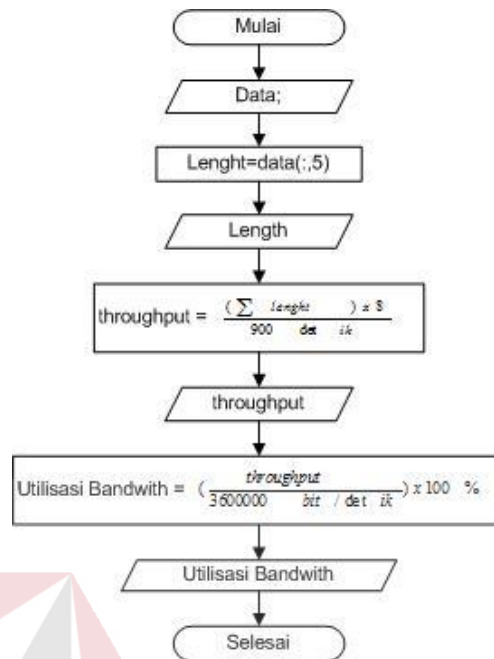
QoS merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan (Yuksel dkk, 2007). QoS (*Quality of Service*) adalah teknologi yang diterapkan pada jaringan WAN (*Wide Area Network*) yang memungkinkan administrator jaringan untuk dapat menangani berbagai efek akibat terjadinya kemacetan (*congestion*) pada lalu lintas aliran paket di dalam jaringan (Yanto, 2013). Perhitungan paket untuk menentukan nilai QoS-nya yaitu Utilisasi *Bandwidth*, *Delay*, *Jitter*, dan *Packet Loss*.

Berikut adalah langkah-langkah yang digunakan untuk menentukan nilai QoS dari paket data yang ter-*capture*.

3.5.1. Utilisasi *Bandwidth*

Pada *flowchart* gambar 3.5 dijelaskan bahwa yang pertama kali dilakukan untuk menghitung Utilisasi *Bandwidth* yaitu pada *MatLab*, file yang telah ter-*filter* IP-nya, dibagi menjadi 2 bagian yaitu matriks *textdata* dan matriks *data*. Matriks *textdata* berisi data yang bertipe *integer*, sedangkan matriks *data* berisi data yang bertipe *double*. Kedua data dipisahkan menjadi 2 matriks dikarenakan data pada matriks *data* akan digunakan untuk perhitungan, sedangkan data pada

matriks *testdata* hanya digunakan untuk mengetahui letak data yang dimaksud pada matriks *data*.

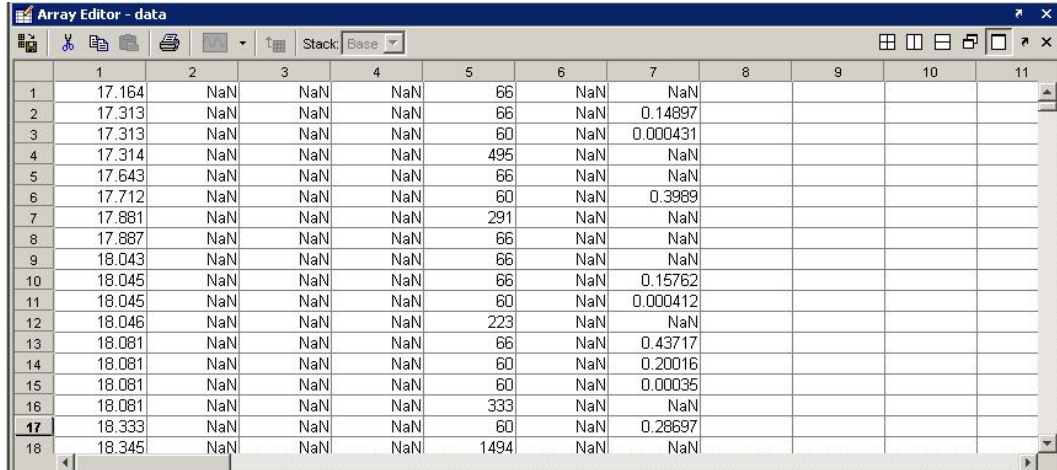


Gambar 3.5. Flowchart perhitungan Utilisasi Bandwidth

Berikut Gambar 3.6 yang menggambarkan bentuk matriks *testdata* dan pada Gambar 3.7 yang menggambarkan bentuk matriks *data* pada *MatLab*.

1	2	3	4	5	6	7	8	9
No.	Time	Source	Destination	Protocol	Length	This is an...	The RTT t...	Message
'183'	"	'192.168.0...	'31.13.79.7'	'TCP'	"	"	"	'Connectio...
'184'	"	'31.13.79.7'	'192.168.0...	'TCP'	"	'183'	"	'Connectio...
'185'	"	'192.168.0...	'31.13.79.7'	'TCP'	"	'184'	"	"
'186'	"	'192.168.0...	'31.13.79.7'	'HTTP'	"	"	"	'GET / HT...
'188'	"	'192.168.0...	'173.252.1...	'TCP'	"	"	"	'Connectio...
'189'	"	'31.13.79.7'	'192.168.0...	'TCP'	"	'186'	"	"
'190'	"	'31.13.79.7'	'192.168.0...	'HTTP'	"	"	"	'HTTP/1.1 ...
'191'	"	'192.168.0...	'31.13.79.7'	'TCP'	"	"	"	'Connectio...
'194'	"	'192.168.0...	'173.252.1...	'TCP'	"	"	"	'Connectio...
'195'	"	'31.13.79.7'	'192.168.0...	'TCP'	"	'191'	"	'Connectio...
'196'	"	'192.168.0...	'31.13.79.7'	'TCP'	"	'195'	"	"
'197'	"	'192.168.0...	'31.13.79.7'	'TLSv1'	"	"	"	"
'198'	"	'173.252.1...	'192.168.0...	'TCP'	"	'188'	"	'Connectio...
'199'	"	'192.168.0...	'31.13.79.7'	'TCP'	"	'190'	"	"
'200'	"	'192.168.0...	'173.252.1...	'TCP'	"	'198'	"	"
'201'	"	'192.168.0...	'173.252.1...	'HTTP'	"	"	"	'GET / HT...
'203'	"	'31.13.79.7'	'192.168.0...	'TCP'	"	'197'	"	"

Gambar 3.6 Tampilan matriks *testdata* pada *MatLab*



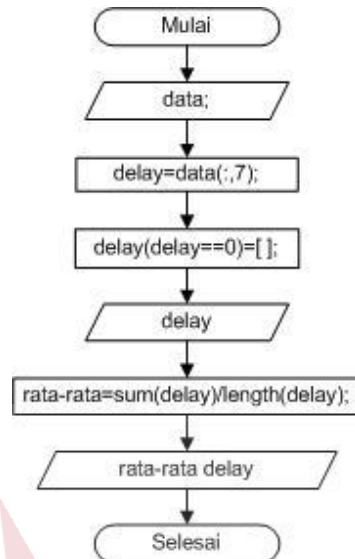
	1	2	3	4	5	6	7	8	9	10	11
1	17.164	NaN	NaN	NaN	66	NaN	NaN				
2	17.313	NaN	NaN	NaN	66	NaN	0.14897				
3	17.313	NaN	NaN	NaN	60	NaN	0.000431				
4	17.314	NaN	NaN	NaN	495	NaN	NaN				
5	17.643	NaN	NaN	NaN	66	NaN	NaN				
6	17.712	NaN	NaN	NaN	60	NaN	0.3989				
7	17.881	NaN	NaN	NaN	291	NaN	NaN				
8	17.887	NaN	NaN	NaN	66	NaN	NaN				
9	18.043	NaN	NaN	NaN	66	NaN	NaN				
10	18.045	NaN	NaN	NaN	66	NaN	0.15762				
11	18.045	NaN	NaN	NaN	60	NaN	0.000412				
12	18.046	NaN	NaN	NaN	223	NaN	NaN				
13	18.081	NaN	NaN	NaN	66	NaN	0.43717				
14	18.081	NaN	NaN	NaN	60	NaN	0.20016				
15	18.081	NaN	NaN	NaN	60	NaN	0.00035				
16	18.081	NaN	NaN	NaN	333	NaN	NaN				
17	18.333	NaN	NaN	NaN	60	NaN	0.28697				
18	18.345	NaN	NaN	NaN	1494	NaN	NaN				

Gambar 3.7. Tampilan matriks *data* pada *MatLab*

Selanjutnya mengambil semua baris pada kolom 5 dari matriks *data*. Kolom 5 dari matriks *data* berjudul *length*, yang berisi besar nilai *byte* masing-masing paket. Kemudian semua nilai pada kolom 5 dijumlahkan, setelah dijumlahkan di kali dengan 8 karena nilainya harus diubah ke satuan *bit* (1 *byte* = 8 *bit*). Nilai yang didapatkan adalah besaran paket per satuan waktu, nilai waktu disini yaitu menggunakan waktu lamanya pengambilan data menggunakan *Wireshark* selama 15 menit. Nilai waktu dikonversikan menjadi satuannya per detik, maka hasil yang telah di kali 8 tadi di bagi 900 detik (15 menit = 900 detik), sehingga didapatkan nilai *throughput*. Setelah diketahui nilai *throughput*-nya, maka dapat dihitung utilisasi *bandwidth*-nya dengan menggunakan rumus perhitungan utilisasi *bandwidth*, yaitu nilai *throughput* dibagi dengan nilai *bandwidth* yang digunakan sebesar 3.6 Mbps (*Mega bit per second*) di konversi menjadi 3.600.000 bps (*bit per second*), kemudian di kali dengan 100 persen. *Bandwidth* yang digunakan pada saat pengambilan data adalah sebesar 3.6 Mbps yang telah disediakan oleh *provider* internet 3 (*three*) dengan sinyal HSDPA.

3.5.2. Delay

Pada *flowchart* Gambar 3.8 merupakan cara untuk menentukan parameter *delay* yang terjadi pada saat pengambilan data.

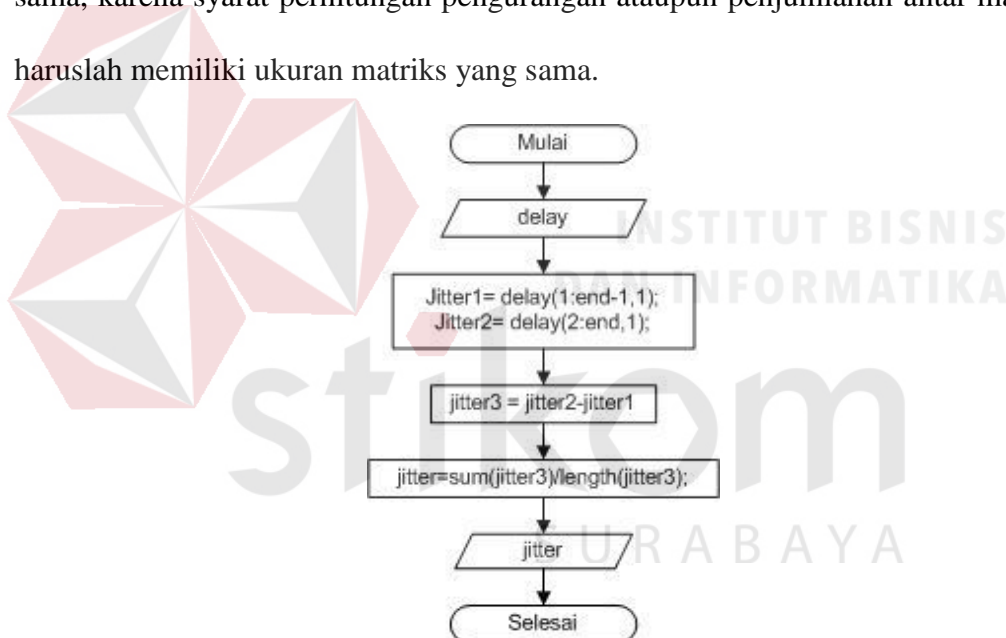


Gambar 3.8. *Flowchart* menentukan *delay*

Pertama kali yang dilakukan yaitu mengambil semua baris pada kolom 7 dari matriks *data*, yang berjudul ‘*The RTT to ACK the segment was*’ RTT (*round trip time*), dengan kata lain merupakan kolom yang berisi sebuah selisih waktu antara segmen yang berisi ACK dari pengiriman data sebelumnya dengan waktu dari segmen yang mengirimkan data. Setelah semua isi pada kolom 7 dari matriks *data* diambil dan dijadikan matriks *delay*, kemudian dilakukan penghapusan baris dari matriks *delay* yang bernilai NaN. Setelah matriks *delay* telah dihapus baris yang bernilai NaN dan didapatkan semua barisnya, maka dilakukan perhitungan rata-ratanya. Cara menghitung rata-ratanya adalah dengan menjumlahkan semua isi dari matriks *delay*, selanjutnya dibagi dengan banyaknya baris dari matriks *delay*. Barulah didapatkan nilai rata-rata *delay* yang dibutuhkan untuk analisis. Hasil dari perhitungan ini adalah sebuah nilai yang memiliki satuan waktu yaitu detik.

3.5.3. Jitter

Pada *flowchart* Gambar 3.9, dapat dilihat bahwa hasil matriks *delay* yang kemudian dibuat 2 buah matriks *jitter*. Dibuatnya 2 buah matriks *jitter* bertujuan agar dalam perhitungan selisih antar *delay* dapat dilakukan dengan tepat karena menggunakan perhitungan matriks. Masing-masing matriks hanya memiliki 1 kolom. Untuk matriks *jitter1* isi dari matriks *delay* pada baris terakhir dihilangkan, sedangkan untuk matriks *jitter2* isi dari matriks *delay* diambil dari baris 2 sampai dengan terakhir. Hal ini dilakukan agar ukuran kedua matriks sama, karena syarat perhitungan pengurangan ataupun penjumlahan antar matriks haruslah memiliki ukuran matriks yang sama.

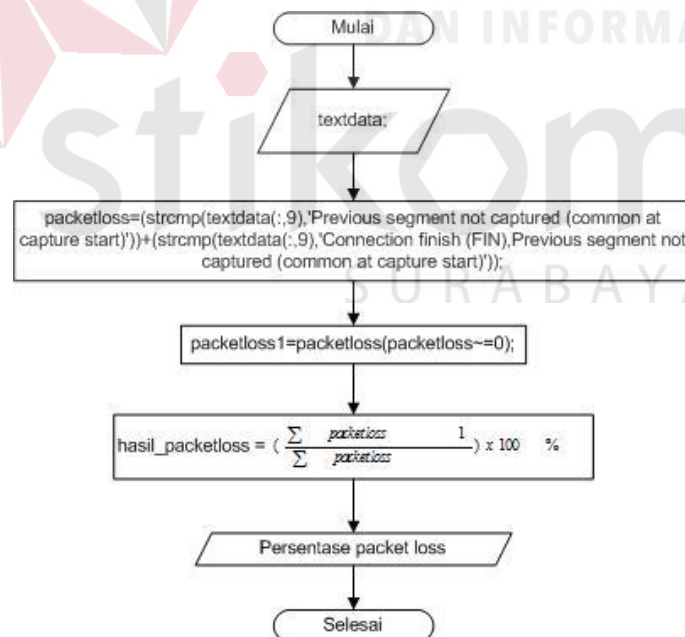


Gambar 3.9. *Flowchart* perhitungan *jitter*

Hasil dari pengurangan matriks *jitter2* dengan matriks *jitter1* bernilai *negatif*, maka hasilnya dibuat nilai mutlak. Kemudian hasilnya dimasukkan dalam rumus perhitungan *jitter* yaitu dengan menjumlahkan semua isi dari matriks *jitter3*, selanjutnya dibagi dengan banyaknya baris dari matriks *jitter3*.

3.5.4. Packet Loss

Pada *flowchart* Gambar 3.10 dijelaskan bahwa *strcmp* difungsikan untuk membandingkan dua string dan akan menghasilkan bilangan bulat, untuk mencari paket data yang hilang dapat dilihat dari sisi kolom 9 pada matriks *textdata* yang berjudul *message*. Kolom ini menerangkan berbagai macam pesan tentang informasi yang lebih spesifik dan sebuah peringatan. Jika pada kolom 9 ini ada baris yang berisikan keterangan ‘*Previous segment not captured (common at capture start)*’ atau ‘*Connection finish (FIN), Previous segment not captured (common at capture start)*’ berarti paket tersebut hilang pada segmen sebelumnya. Setelah itu dilakukan pengecekan kembali di baris berikutnya sampai jumlah batas terakhir dari matriks *textdata*. Jika sudah terkumpulkan, maka isi dari matriks dapat diketahui jumlah paket yang hilang.



Gambar 3.10. *Flowchart* perhitungan *packet loss*

Dari sini maka dilakukan perhitungan prosentase *packet loss* dengan menggunakan rumus perhitungan *packet loss* yaitu hasil paket yang hilang dibagi

dengan jumlah paket yang ada, kemudian dikalikan 100 persen. Hasil dari perhitungan ini adalah sebuah prosentase *packet loss*.

3.6. Evaluasi Sistem

Setelah memperoleh nilai dari parameter-parameter QoS-nya, maka dapat dilakukan analisis. Analisis yang dilakukan yaitu menganalisis nilai parameter-parameter QoS yang didapat, kemudian barulah diketahui karakteristik lalu lintas data aplikasi *web social networking* yaitu *facebook* dan *twitter*. Karakteristik lalu lintas data yang dimaksud adalah dilihat dari hasil perhitungan parameter QoS dari arsitektur jaringan yang dibuat, kemudian dibandingkan dengan nilai parameter QoS yang sesuai dengan standar yang ada. Dengan begitu dapat mengetahui bagaimana kinerja dari arsitektur jaringan yang dibuat, serta karakteristik dari lalu lintas data dari aplikasi *web social network facebook* dan *twitter*.

