

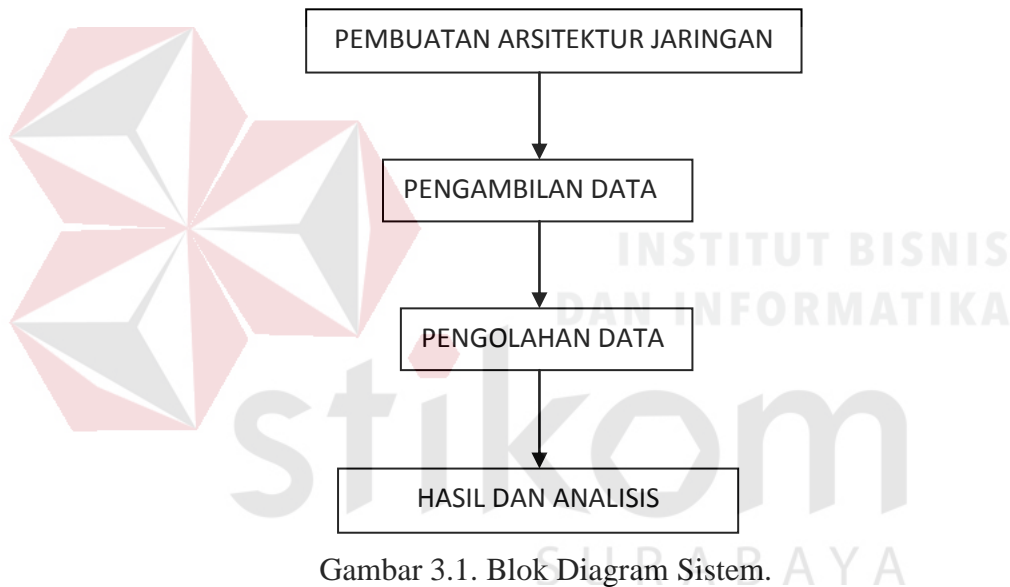
BAB III

METODOLOGI PENELITIAN

Pada bab ini membahas tentang perancangan blok diagram sistem, arsitektur jaringan, pengambilan data, pengolahan data, dan analisis data.

3.1. Perancangan Sistem dan Blok Diagram Sistem

Perancangan sistem dapat dijelaskan dengan lebih baik melalui blok diagram seperti yang terlihat seperti Gambar 3.1.

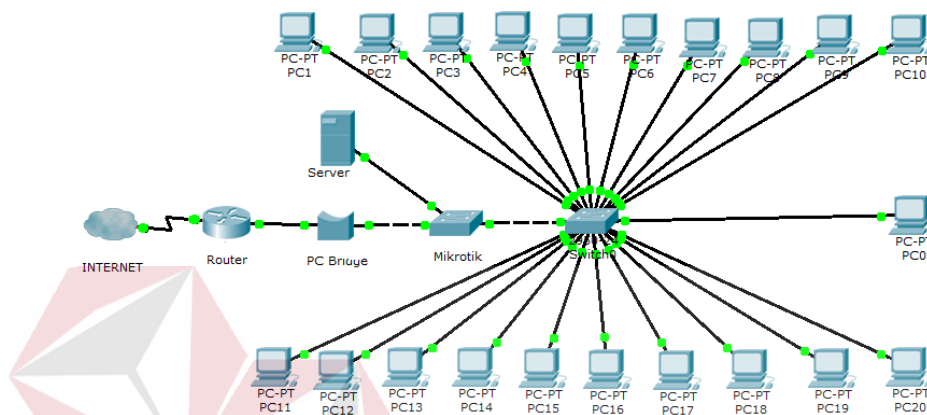


Gambar 3.1. Blok Diagram Sistem.

Dari Gambar 3.1 blok diagram sistem dapat diketahui cara kerja dari keseluruhan penelitian yang dilakukan. Mulai dari penyesuaian arsitektur jaringan pada game center, cara pengambilan data, dan cara pengolahan data untuk mendapatkan hasil dari perhitungan parameter-parameter *Quality of Service* (QoS) yaitu *Delay*, *Jitter*, *Packet Loss*, dan *utilisasi Bandwidth*, serta analisis dari parameter QoS dan analisis parameter statistik.

3.2. Arsitektur Jaringan

Arsitektur jaringan pada *game center* disesuaikan sedemikian rupa agar dapat membantu proses pengambilan data. Arsitektur jaringan yang dibuat dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2. Arsitektur Jaringan.

Komponen dari arsitektur jaringan yang dibuat adalah :

1. Sebuah Modem Router menggunakan Mediaplay sebagai layanan *provider* internet.
2. Sebuah PC *Server*.
3. Sebuah PC *Bridge*.
4. Sebuah Mikrotik.
5. Sebuah *Switch*.
6. 21 Buah PC Client. Masing-masing memiliki IP 192.168.5.1 sampai IP 192.168.5.21 seperti terlihat pada Gambar 3.2.

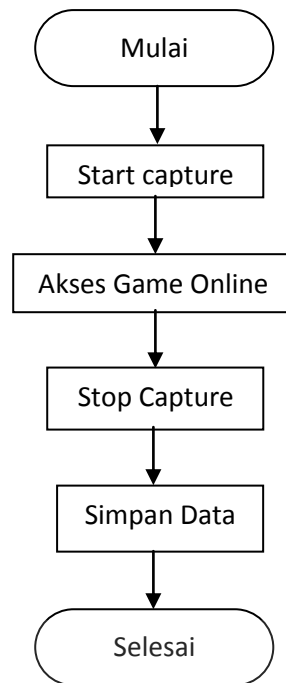
Game center menggunakan *provider* internet dengan kecepatan hingga 10 Mbps. Untuk memudahkan pengambilan data router dihubungkan ke PC *Bridge*. PC *Bridge* adalah sebuah PC yang memiliki 2 buah *interface* yaitu 2 buah

Network Interface Card (NIC). Satu *interface* terhubung dengan mikrotik dan *interface* yang lain terhubung ke modem router. Mikrotik terhubung langsung ke *switch* yang menjadi penghubung antara PC *client* dan mikrotik. Mikrotik juga terhubung dengan sebuah PC *server*. Pada PC *Bridge* dibentuk sebuah *bridge* melalui dua *interface* yang ada. Setiap paket data yang keluar dari mikrotik dan menuju ke modem router akan melewati PC bridge. Jadi setiap paket data game online yang diakses oleh PC *client* dapat dilihat melalui PC *bridge*. Untuk menangkap paket data yang melintas pada PC *bridge* digunakan aplikasi *Network Protocol Analyzer* yaitu. *Wireshark* digunakan karena merupakan *Network Protocol Analyzer* yang dapat digunakan pada *Operating System (OS) Windows*. OS yang digunakan pada PC *bridge* adalah windows7.

3.3. Pengambilan Data

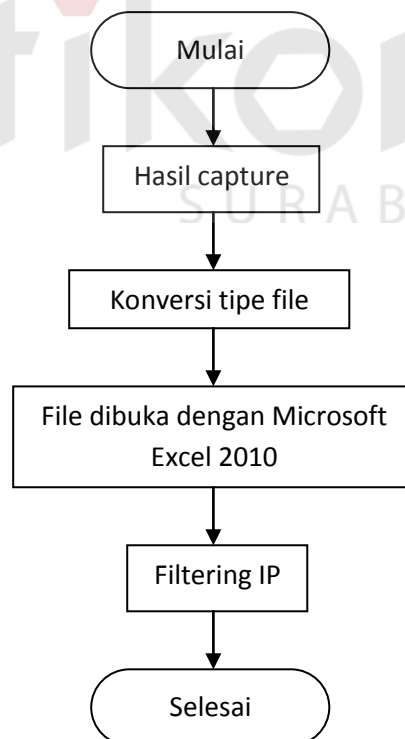
Metode pengambilan paket data adalah dengan menggunakan *Network Protocol Analyser* yaitu *Wireshark* yang telah di-*install* di komputer bridge. Komputer bridge menghubungkan router dan mikrotik, yang dimana router terhubung langsung ke internet sedangkan mikrotik terhubung ke *switch* yang menghubungkan komputer-komputer *client*.

Dari gambar 3.3 dapat dijelaskan *Network Protocol Analyzer* mengambil paket data pada *interface* komputer *bridge*, setiap paket data yang melintas keluar dan masuk dalam jaringan dapat dilihat melalui *Wireshark*. Setelah proses pengambilan data selesai, data selanjutnya disimpan dan siap untuk diolah.



Gambar 3.3. Blok diagram pengambilan data.

3.4. Pengolahan Data



Gambar 3.4. Blok diagram pengolahan data.

Dari Gambar 3.4 dapat dijelaskan data yang telah di-*capture* menggunakan *Wireshark* disimpan dalam *format file* yang hanya bisa dibaca oleh *Wireshark*. Agar data tersebut dapat diolah, data harus dikonversi ke bentuk file yang dapat dibaca aplikasi lain, contohnya adalah *Microsoft Excel*.

Setelah melakukan konversi pada file dan data dapat dibaca oleh *Microsoft Excel*. Selanjutnya melakukan *filtering* IP, dalam proses analisis hanya diperlukan IP yang berhubungan dengan *game online*. Kemudian data dapat digunakan untuk proses analisis.

Tabel 3.1 IP address aplikasi *game online*

Nama Owner/Publisher	IP address (Net Range)	Nama Game
Pt. Kreon	203.89.146.0 - 203.89.147.255 49.50.4.0 - 49.50.7.255	Dragon Nest INA Lost Saga Point Blank
Valve Corp.	103.28.56.0 - 103.28.55.255 72.165.61.0 - 72.165.61.255 146.66.153.0 - 146.66.157.255 185.25.182.0 - 185.25.183.255 208.64.200.0 - 208.64.203.255 208.78.164.0 - 208.78.164.255	Dota 2
Cherry Credits Pte Ltd	203.166.155.0 - 203.166.155.255	Dragon Nest SEA
Garena Online Pte Ltd	203.116.155.0 - 203.116.155.255 203.117.155.0 - 203.117.155.255 203.117.172.0 - 203.117.172.255	Heroes of Newerth
Pt. CJ Internet Indonesia	103.249.59.0 - 103.249.59.255 182.16.248.0 - 182.16.248.255	Elsword Special Force 2

Berikut tabel 3.1 adalah daftar IP dari aplikasi game online yang didapat pada proses pengambilan data. Cara untuk mengetahui IP aplikasi *game online* yang akan digunakan untuk analisis adalah dengan memasukkan IP yang di tangkap oleh *Wireshark* ke aplikasi *web* pencari IP Address/ Domain Name Lookup, yang beralamatkan <http://cqcounter.com/whois>.

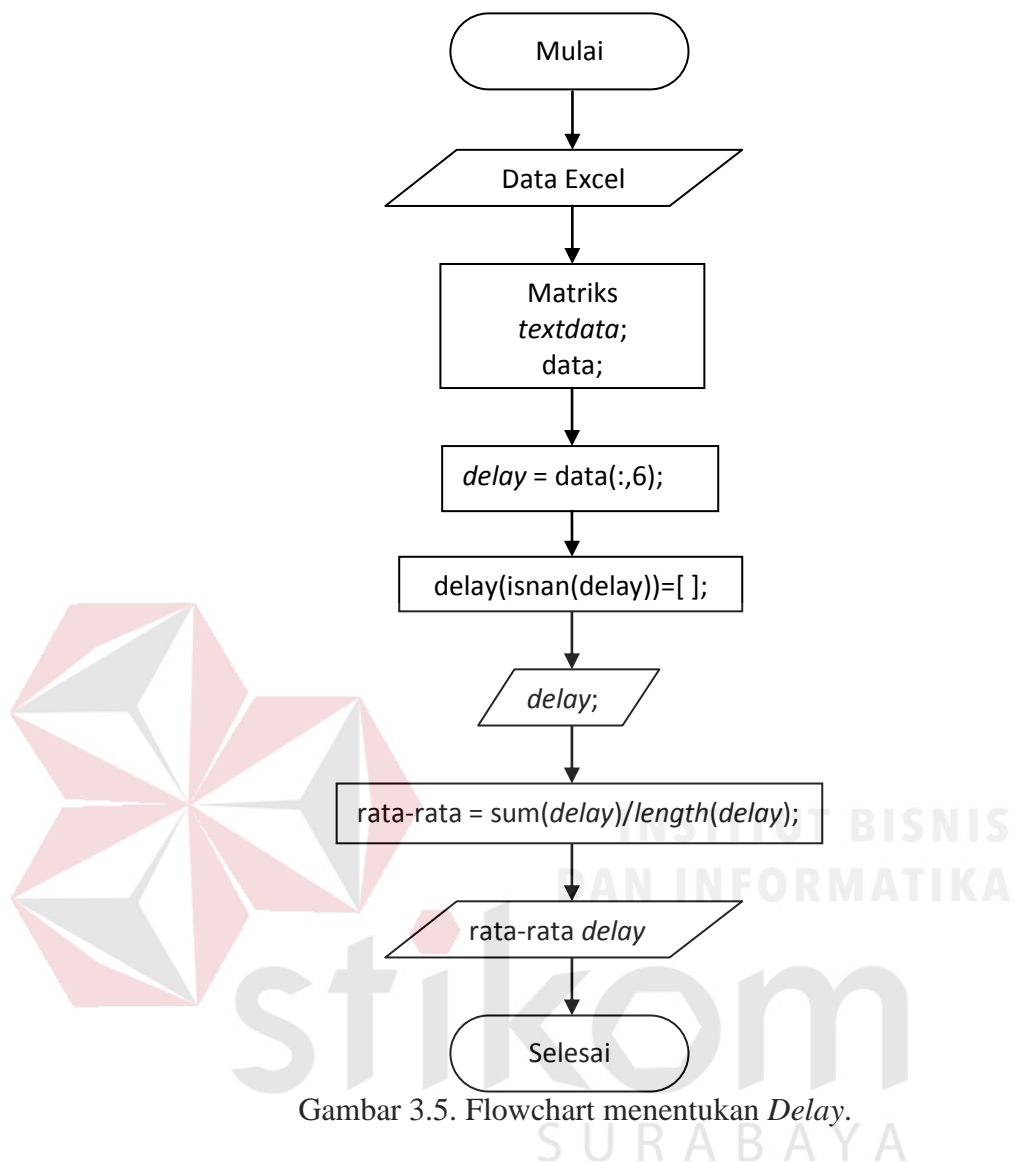
3.5. Analisis Data

Quality of Service (QoS) didefinisikan mekanisme jaringan yang memungkinkan aplikasi-aplikasi atau layanan dapat beroperasi sesuai dengan yang diharapkan. Melalui QoS, seorang *network administrator* dapat mengambil keputusan untuk menyelesaikan masalah pada trafik jaringannya. Dengan implementasi analisis parameter QoS di jaringan ini, diharapkan dapat mengetahui sumber masalah yang terjadi di trafik pada jaringan, serta dapat menganalisis karakteristiknya.

Perhitungan untuk analisis karakteristik paket data adalah dari sisi *Delay*, *Jitter*, *Packet Loss*, dan Utilisasi *Bandwidth*. Berikut adalah langkah-langkah yang digunakan untuk menentukan nilai QoS dari paket data.

3.5.1. Delay

Flowchart Gambar 3.5 merupakan cara untuk menghitung *delay* yang terjadi pada jaringan saat proses data diambil. Pada Matlab, file yang telah di-*filter* IP-nya, dibagi menjadi 2 bagian, 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 *textdata* hanya digunakan untuk mengetahui letak data yang dimaksud pada matriks data.



Berikut Gambar 3.6 yang menggambarkan bentuk matriks *textdata* pada *MatLab*.

No.	Time	Source	Destination	Protocol	Length	The RTT to ...	This is an A...	Message
1	0	203.89.146....	192.168.1.2	UDP	233			
2	5.2400e-04	192.168.1.2	122.102.53.27	UDP	96			
3	0.0022	103.248.58.58	192.168.1.2	UDP	188			
4	0.0028	192.168.1.2	103.248.58.58	UDP	60			
5	0.0176	203.89.146....	192.168.1.2	UDP	252			
6	0.0204	192.168.1.2	122.102.53.2	TCP	61			
7	0.0242	192.168.1.2	103.249.58....	UDP	60			
8	0.0280	192.168.1.2	122.102.53.28	UDP	94			
9	0.0283	122.102.53.28	192.168.1.2	UDP	330			
10	0.0296	192.168.1.2	122.102.49.75	TCP	96			
11	0.0309	192.168.1.2	122.102.53.26	UDP	92			
12	0.0321	192.168.1.2	122.102.53.28	UDP	91			
13	0.0322	103.248.58.58	192.168.1.2	UDP	68			
14	0.0327	192.168.1.2	103.248.58.58	UDP	60			
15	0.0397	49.50.4.19	192.168.1.2	UDP	85			
16	0.0397	49.50.4.19	192.168.1.2	UDP	85			
17	0.0400	122.102.53.27	192.168.1.2	UDP	222			
18	0.0407	192.168.1.2	122.102.53.27	UDP	101			
19	0.0426	203.89.147....	192.168.1.2	UDP	78			
20	0.0454	122.102.49.75	192.168.1.2	TCP	110			
21	0.0483	122.102.49.75	192.168.1.2	TCP	110	0.0187	10	
22	0.0572	122.102.53.28	192.168.1.2	UDP	344			
23	0.0587	192.168.1.2	203.89.147....	UDP	87			
24	0.0608	203.89.147....	192.168.1.2	UDP	82			

Gambar 3.6. Tampilan matriks *textdata* pada *MatLab*.

Langkah pertama adalah menggunakan seluruh elemen pada Kolom 6 yang berjudul 'The RTT to ACK the segment was' untuk menghitung delay. Kolom 6 berisi selisih waktu antara segmen yang berisi ACK dari pengiriman data sebelumnya dengan waktu dari segmen yang mengirimkan data. Melakukan import pada data sehingga mendapatkan matriks *delay*. Gambar 3.7 menggambarkan matriks data yang memiliki 6 kolom pada *MatLab*.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	NaN	NaN	NaN	233	NaN	NaN	NaN				
2	5.2400e-04	NaN	NaN	NaN	96	NaN	NaN	NaN				
3	0.0022	NaN	NaN	NaN	188	NaN	NaN	NaN				
4	0.0028	NaN	NaN	NaN	60	NaN	NaN	NaN				
5	0.0176	NaN	NaN	NaN	252	NaN	NaN	NaN				
6	0.0204	NaN	NaN	NaN	61	NaN	NaN	NaN				
7	0.0242	NaN	NaN	NaN	60	NaN	NaN	NaN				
8	0.0280	NaN	NaN	NaN	94	NaN	NaN	NaN				
9	0.0283	NaN	NaN	NaN	330	NaN	NaN	NaN				
10	0.0296	NaN	NaN	NaN	96	NaN	NaN	NaN				
11	0.0309	NaN	NaN	NaN	92	NaN	NaN	NaN				
12	0.0321	NaN	NaN	NaN	91	NaN	NaN	NaN				
13	0.0322	NaN	NaN	NaN	68	NaN	NaN	NaN				
14	0.0327	NaN	NaN	NaN	60	NaN	NaN	NaN				
15	0.0397	NaN	NaN	NaN	85	NaN	NaN	NaN				
16	0.0397	NaN	NaN	NaN	85	NaN	NaN	NaN				
17	0.0400	NaN	NaN	NaN	222	NaN	NaN	NaN				
18	0.0407	NaN	NaN	NaN	101	NaN	NaN	NaN				
19	0.0426	NaN	NaN	NaN	78	NaN	NaN	NaN				
20	0.0454	NaN	NaN	NaN	110	NaN	NaN	NaN				
21	0.0483	NaN	NaN	NaN	110	0.0187	NaN	NaN				
22	0.0572	NaN	NaN	NaN	344	NaN	NaN	NaN				

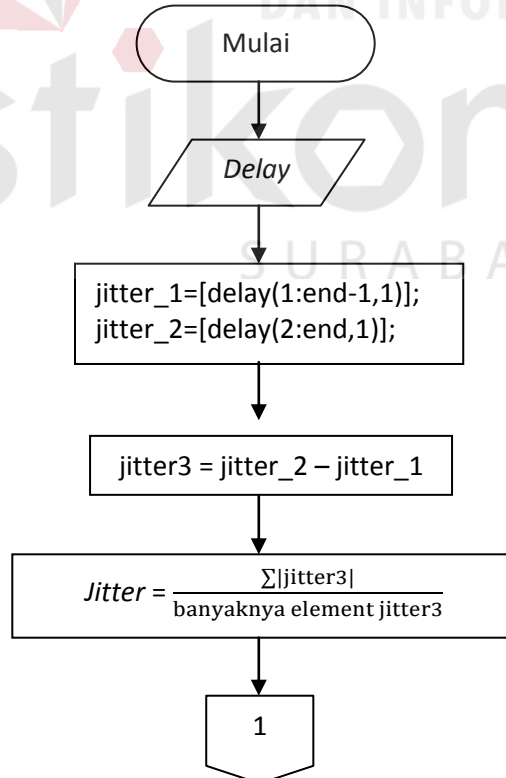
Gambar 3.7. Tampilan matriks data pada *MatLab*.

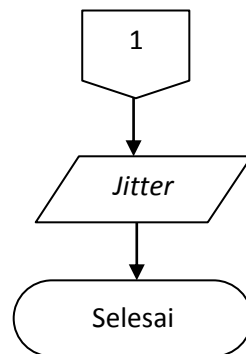
Setelah mendapatkan matriks *delay*, lalu dilakukan proses penghitungan *delay*. Cara menghitung rata-ratanya adalah dengan menjumlahkan semua elemen pada matriks *delay* kemudian membagi hasilnya dengan banyaknya elemen pada

matriks *delay*. Maka didapatkan nilai rata-rata delay yang memiliki satuan detik dan selanjutnya akan digunakan untuk analisis parameter QoS.

3.5.2. Jitter

Pada *flowchart* Gambar 3.8 dapat dilihat, dari hasil perhitungan *delay* dapat dibuat menjadi 2 buah matriks *jitter*. Hal ini bertujuan agar dalam perhitungan selisih antar *delay* dapat dilakukan dengan tepat karena menggunakan perhitungan matriks. Elemen baris pertama pada kolom *jitter1* dihilangkan sedangkan pada kolom *jitter2* elemen pada baris pertama dihilangkan. Matriks kolom *jitter1* dan *jitter2* akan memiliki ukuran yang sama. Setelah itu dilakukan pengurangan antara matriks *jitter2* dengan *jitter1*. Hasil pengurangan diubah kedalam nilai mutlak. Lalu nilai hasil pengurangan tersebut digunakan untuk penghitungan nilai *jitter* yang memiliki satuan detik.

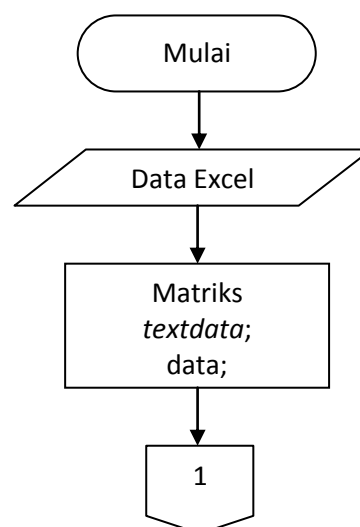


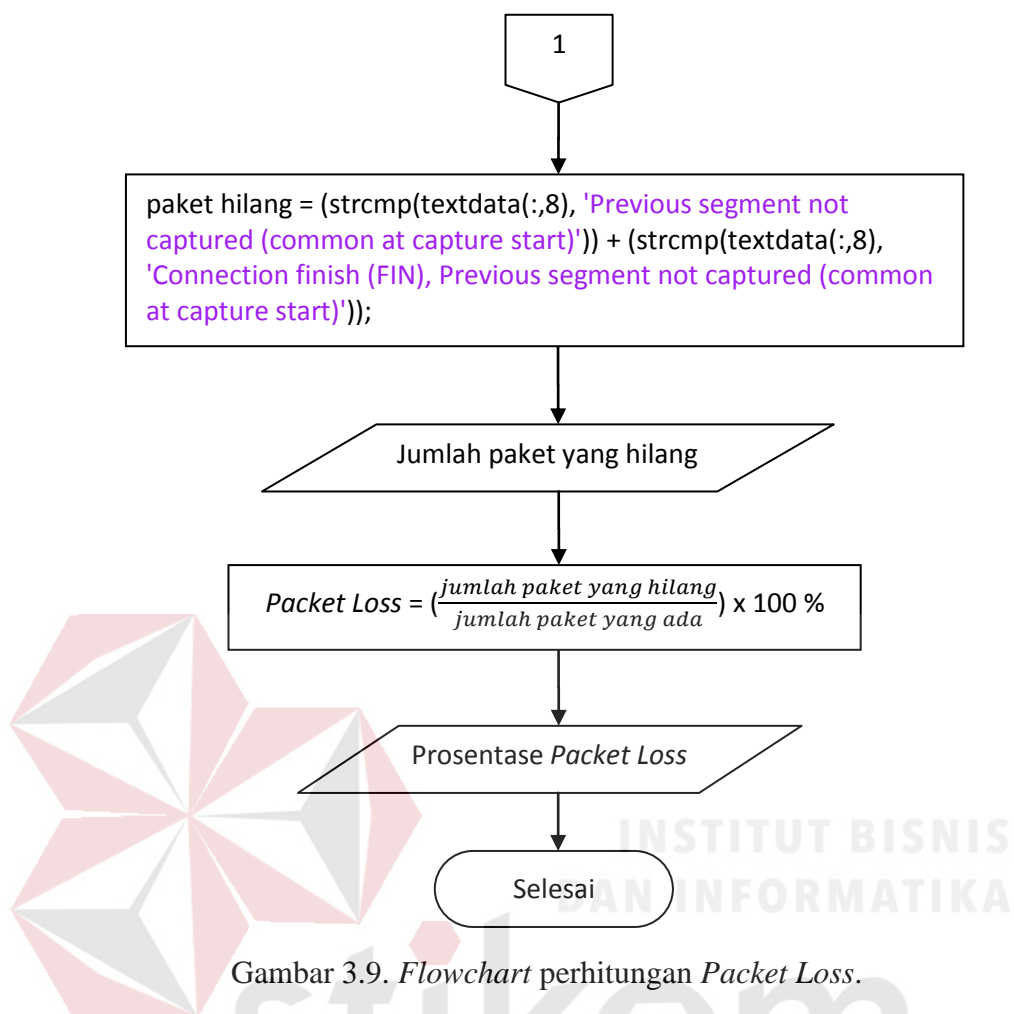


Gambar 3.8. *Flowchart* perhitungan *Jitter*.

3.5.3. *Packet Loss*

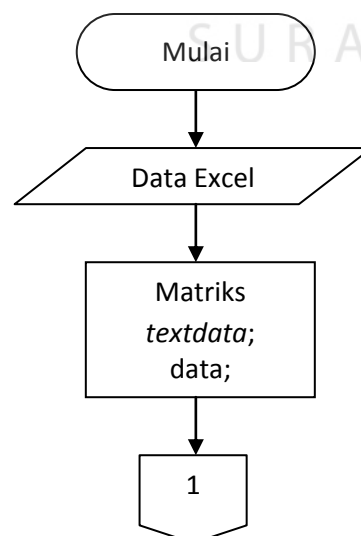
Pada *flowchart* Gambar 3.9 untuk menentukan paket data yang hilang dapat dilihat pada kolom 10 berjudul *message*. Kolom ini menjelaskan tentang informasi spesifik atau sebuah peringatan pada setiap data. Jika pada kolom 10 terdapat keterangan '*Previous segment not captured (common at capture start)*' berarti paket tersebut hilang pada segmen sebelumnya. Lakukan pencocokan isi dari setiap baris pada kolom 10, maka akan ditemukan jumlah paket yang hilang. Selanjutnya gunakan rumus perhitungan *packet loss* untuk mengetahui prosentase *packet loss*.

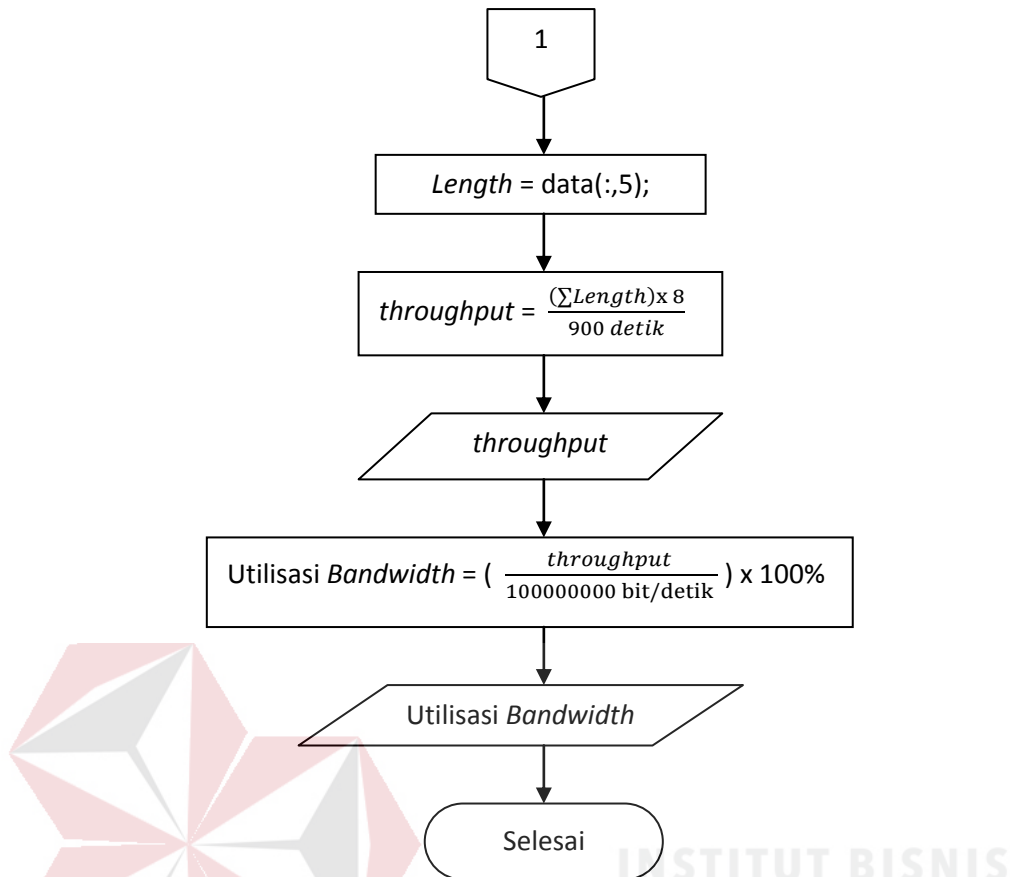




Gambar 3.9. Flowchart perhitungan *Packet Loss*.

3.5.4. Utilisasi *Bandwidth*





Gambar 3.10. *Flowchart* perhitungan *Utilisasi Bandwidth*.

Pada *flowchart* Gambar 3.10 dapat dilihat langkah pertama adalah mengambil semua element pada kolom 5. Kolom 5 berisi besar nilai *byte* masing-masing paket. Kemudian menjumlahkan semua isi dari kolom 5 dan dikalikan dengan 8 untuk mengubah nilai *byte* menjadi bit.

Setelah itu menghitung nilai *throughout* dengan cara membagi nilai hasil dari proses perkalian dengan lamanya waktu pengambilan data dalam satuan detik. Lama pengambilan data adalah 15 menit dan dikonversi menjadi 900 detik.

Selanjutnya untuk mendapatkan prosentase utilisasi bandwidth adalah dengan cara membagi nilai hasil *throughput* dengan besarnya *bandwidth* yang disediakan.

Bandwidth yang digunakan adalah sebesar 10 Mbps. *Bandwidth* tersebut

disediakan oleh *provider* internet MediaPlay, sehingga utilisasi *bandwidth* yang diukur adalah dari *client* ke *workstation* milik provider internet MediaPlay. Besarnya *bandwidth* setelah melewati *workstation* tidak diketahui karena *routing* protokolnya tidak diketahui.

3.4.5 Estimasi Parameter Statistik

Estimasi dilakukan untuk mendapatkan parameter dari setiap distribusi yang mendekati distribusi panjang paket data. Langkah – Langkah analisis data:

1. Membuat tabel distribusi frekuensi dimana terdapat jumlah kelas, interval kelas untuk paket data length, frekuensi dari paket data agar memudahkan data saat disajikan dalam bentuk histogram atau grafik.
2. Selanjutnya melakukan *fitting* distribusi. *Fitting* distribusi bertujuan untuk membandingkan distribusi panjang paket data dengan bentuk-bentuk distribusi data lainnya. Proses *fitting* ini membandingkan pola grafik dari data dengan pola grafik dari jenis distribusi lainnya. Proses *fitting* diawali dengan melakukan *import file* data excel ke matlab. Kemudian menggunakan salah satu fitur pada matlab yaitu *distribution fitting tool* dengan mengetikkan (`>>dfittool`). Kemudian memasukkan data yang telah di-*import* kedalam *fitting* tool dan membandingkan pola grafik data dengan pola grafik distribusi yang ada.
3. Setelah menyelesaikan *fitting* distribusi, lakukan pengamatan visual terhadap hasil *fitting* distribusi untuk menentukan bentuk distribusi yang sesuai.

3.5. Evaluasi Sistem

Setelah memperoleh nilai dari parameter-parameter QoS-nya, maka dapat dilakukan analisis. Analisis yang dilakukan adalah menganalisis nilai parameter-parameter QoS yang didapat dan estimasi parameter statistik yang sesuai, kemudian barulah diketahui karakteristik lalu lintas data aplikasi *game online*. 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 aplikasi *game online*.

