

BAB 3 METODOLOGI

Dalam bab metodologi ini akan dibahas tentang studi literatur, analisis kebutuhan yang membahas kebutuhan fungsional dan kebutuhan simulasi, perancangan simulasi, implementasi simulasi, pengujian, analisis hasil dan kesimpulan dan saran. Gambar 3.1 menyajikan diagram alir tentang metode penelitian yang digunakan.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literatur digunakan untuk mempelajari tentang dasar dari teori-teori yang akan digunakan untuk mendukung penulisan penelitian ini. Dasar dari teori-teori tersebut didapatkan dari beberapa sumber seperti: buku, *e-book*, skripsi, jurnal, artikel dan *website*. Dasar teori yang digunakan untuk mendukung penulisan penelitian ini yaitu *Transmission Control Protokol (TCP)*, *TCP Vegas*, *TCP New Reno*, *Antrian*, *Droptail*, *Random Early Detection*, *Network Simulator 2*, *Topologi Abilene* dan *Quality of Service (QoS)* sebagai parameter hasil diantaranya: *packet delivery ratio*, *throughput*, *delay* dan *packet loss*.

3.2 Analisis Kebutuhan

Analisis kebutuhan memiliki tujuan untuk menganalisa semua kebutuhan yang diperlukan dalam membangun simulasi yang akan dilakukan pada penelitian ini, sehingga nantinya dapat diperoleh kebutuhan yang sesuai untuk simulasi.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi tentang proses apa saja dan hasil yang dapat dilakukan saat simulasi. Pada penelitian ini ada beberapa kebutuhan fungsional pada simulasi, yaitu sebagai berikut:

1. Simulasi dapat menjalankan TCP Vegas dan TCP New Reno dengan manajemen antrian *Random Early Detection* dan *Droptail* menggunakan NS2.
2. Simulasi dapat menghasilkan nilai mentah berupa *file trace* yang nantinya dapat diolah dan dianalisis menjadi parameter hasil.

3.2.2 Kebutuhan Non-Fungsional

Kebutuhan *non-fungsional* merupakan kebutuhan yang diperlukan untuk membangun lingkungan simulasi. Dalam penelitian ini, kebutuhan tersebut terbagi menjadi dua yaitu kebutuhan simulasi dan kebutuhan jaringan, yang dapat disebutkan sebagai berikut:

3.2.2.1 Kebutuhan Simulasi

Kebutuhan simulasi yaitu kebutuhan yang digunakan sebagai tempat untuk merancang dan menjalankan simulasi dalam penelitian ini. Kebutuhan simulasi dibagi dua yaitu perangkat keras dan perangkat lunak. Di bawah ini merupakan kebutuhan simulasi yang digunakan, yaitu sebagai berikut:

1. Perangkat Keras
 - Laptop Lenovo G40-70
2. Perangkat Lunak
 - Sistem Operasi Ubuntu 16.04 64 bit
 - *Network Simulator* 2.35
 - *Network Animator* (NAM)

3.2.2.2 Kebutuhan Jaringan

Kebutuhan jaringan yaitu kebutuhan untuk membangun topologi jaringan yang digunakan dalam penelitian ini. Ada beberapa komponen yang dibutuhkan untuk membangun topologi jaringan dalam penelitian ini. Di bawah ini merupakan kebutuhan jaringan yang digunakan, yaitu sebagai berikut:

1. *Router*
2. *Node Sumber*
3. *Node Tujuan*
4. *Link*
5. *Buffer*

3.3 Perancangan Simulasi

Perancangan merupakan tahap setelah analisis kebutuhan, dimana pada tahap ini menjelaskan mekanisme dari simulasi yang akan dilakukan. Tahap perancangan dilakukan agar simulasi terstruktur dan terarah. Perancangan simulasi memiliki beberapa tahap yang harus dilakukan, yaitu sebagai berikut:

1. Analisis Kebutuhan

Menjelaskan semua kebutuhan yang digunakan untuk membangun lingkungan simulasi. Dimana kebutuhan tersebut dibagi menjadi dua yaitu kebutuhan simulasi dan kebutuhan jaringan.

2. Alur Kerja Simulasi

Menjelaskan tentang alur kerja dari simulasi mulai awal menentukan varian TCP dan manajemen antrian lalu proses pengiriman data sampai terjadi *congestion*.

3. Perancangan Simulasi

Menjelaskan parameter-parameter apa saja yang akan digunakan dalam simulasi pada penelitian ini.

4. Perancangan Topologi

Menentukan topologi jaringan yang akan dibangun untuk simulasi. Berapa jumlah *node* dan apa peran *node* dalam topologi tersebut.

5. Perancangan TCP Vegas

Menjelaskan bagaimana mekanisme TCP Vegas dalam mengirimkan data dari sumber ke tujuan ketika jaringan terjadi *congestion*.

6. Perancangan TCP New Reno

Menjelaskan bagaimana mekanisme TCP New Reno dalam mengirimkan data dari sumber ke tujuan ketika jaringan terjadi *congestion*.

7. Perancangan *Random Early Detection*

Menjelaskan bagaimana mekanisme *Random Early Detection* dalam melayani data yang masuk ke dalam antrian sampai terjadi kondisi dimana antrian diharuskan melakukan *packet drop*.

8. Perancangan *Droptail*

Menjelaskan bagaimana mekanisme *Droptail* dalam melayani data yang masuk ke dalam antrian sampai terjadi kondisi dimana antrian diharuskan melakukan *packet drop*.

3.4 Implementasi Simulasi

Pada penelitian ini, implementasi dimulai dengan melakukan proses instalasi *Network Simulator 2.35* pada ubuntu 16.04. Setelah *Network Simulator 2.35* berhasil ter-*install*, dilanjutkan dengan membangun lingkungan simulasi dengan konfigurasi *script* berekstensi *.tcl*. Dimana dalam *script* tersebut berisi implementasi pembuatan topologi jaringan yang telah dirancang sebelumnya, menerapkan varian TCP dan *traffic source* yang digunakan, mengatur manajemen antrian apa yang digunakan, menerapkan besar paket data yang akan dikirim, lama waktu simulasi serta besar *bandwidth* dan *delay* setiap *link*. Tahap pengujian nantinya akan sangat bergantung pada tahap perancangan dan implementasi.

3.5 Pengujian

Pada penelitian ini, pengujian dilakukan untuk menunjukkan bahwa simulasi dapat berjalan sesuai dengan implementasi konfigurasi *script* yang telah dibuat sebelumnya atau tidak. Pengujian ini akan membandingkan TCP Vegas dan TCP New Reno dengan menambahkan antrian *Random Early Detection* dan *Droptail* untuk mengatasi *congestion* pada NS-2 dengan mengamati *packet delivery ratio*, *throughput*, *delay* dan *packet loss*. Dimana terdapat dua pengujian yang akan dilakukan yaitu sebagai berikut:

3.5.1 Skenarion Pengujian Pertama Varian TCP Dengan *Droptail*

Pada skenario pengujian pertama akan dilakukan pengujian dengan menerapkan TCP Vegas dan TCP New Reno menggunakan antrian *Droptail* dengan skema penambahan kapasitas *buffer* yaitu 20, 30, 40, 50 dan 60 paket. *Traffic source* yang digunakan yaitu FTP dengan *packet size* 1024 bytes dan lama waktu simulasi 300 detik. Kemudian akan dilihat perbandingan kinerjanya dengan melihat parameter yang telah ditentukan.

3.5.1.1 Pengujian *Packet Delivery Ratio*

Pengujian Packet Delivery Ratio dilakukan untuk melihat perbandingan jumlah paket yang diterima oleh *node* tujuan dan jumlah paket yang dikirimkan oleh *node* sumber dalam kurun waktu tertentu. Pengujian dilakukan dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *packet delivery ratio*.

3.5.1.2 Pengujian *Throughput*

Pengujian *Throughput* dilakukan untuk melihat banyaknya paket yang tiba atau diterima dalam interval waktu yang ditentukan. Pengujian dilakukan dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *throughput*.

3.5.1.3 Pengujian *Delay*

Pengujian *Delay* dilakukan untuk melihat waktu yang dibutuhkan suatu paket saat dikirimkan dari *node* sumber ke *node* tujuan. Pengujian dilakukan

dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *delay*.

3.5.1.4 Pengujian *Packet Drop*

Pengujian *Packet drop* merupakan paket yang terbuang saat proses transmisi dari *node* sumber ke tujuan. Pengujian dilakukan dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *packet drop*.

3.5.2 Skenario Pengujian Kedua Varian TCP Dengan *Random Early Detection*

Pada skenario pengujian kedua akan dilakukan pengujian dengan menerapkan TCP Vegas dan TCP New Reno menggunakan antrian *Random Early Detection*. Dimana terdapat dua skema yang dilakukan, yang pertama dengan skema penambahan *min thresh* yaitu 20, 25, 30, 35 dan 40 paket dengan *max thresh* 50 paket dan untuk yang kedua dengan skema penambahan *max thresh* yaitu 30, 35, 40, 45 dan 50 paket dengan *min thresh* 20 paket. Untuk kapasitas *buffer* pada antrian akan tetap yaitu 60 paket. *Traffic source* yang digunakan yaitu FTP dengan *packet size* 1024 *bytes* dan lama waktu simulasi 300 detik. Kemudian akan dilihat perbandingan kinerjanya dengan melihat parameter yang telah ditentukan.

3.5.2.1 Pengujian *Packet Delivery Ratio*

Pengujian Packet Delivery Ratio dilakukan untuk melihat perbandingan jumlah paket yang diterima oleh *node* tujuan dan jumlah paket yang dikirimkan oleh *node* sumber dalam kurun waktu tertentu. Pengujian dilakukan dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *packet delivery ratio*.

3.5.2.2 Pengujian *Throughput*

Pengujian Throughput dilakukan untuk melihat banyaknya paket yang tiba atau diterima dalam interval waktu yang ditentukan. Pengujian dilakukan dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *throughput*.

3.5.2.3 Pengujian *Delay*

Pengujian Delay dilakukan untuk melihat waktu yang dibutuhkan suatu paket saat dikirimkan dari *node* sumber ke *node* tujuan. Pengujian dilakukan dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *delay*.

3.5.2.4 Pengujian *Packet Drop*

Pengujian Packet drop merupakan paket yang terbuang saat proses transmisi dari *node* sumber ke tujuan. Pengujian dilakukan dengan mengolah *file trace* dengan *script awk* yang telah dibuat untuk menghitung *packet drop*.

Proses mengolah *file trace* dengan ekstensi *.tr* dari hasil pengujian dapat dilakukan dengan *awk script*. Contoh *file trace* dapat dilihat pada Gambar 3.2.

```
+ 0.2 0 2 tcp 40 ----- 1 0.0 4.0 0 0
- 0.2 0 2 tcp 40 ----- 1 0.0 4.0 0 0
r 0.20832 0 2 tcp 40 ----- 1 0.0 4.0 0 0
+ 0.20832 2 3 tcp 40 ----- 1 0.0 4.0 0 0
- 0.20832 2 3 tcp 40 ----- 1 0.0 4.0 0 0
r 0.3096 2 3 tcp 40 ----- 1 0.0 4.0 0 0
+ 0.3096 3 4 tcp 40 ----- 1 0.0 4.0 0 0
- 0.3096 3 4 tcp 40 ----- 1 0.0 4.0 0 0
r 0.360133 3 4 tcp 40 ----- 1 0.0 4.0 0 0
```

Gambar 3.2 Contoh File Trace

Setelah *file trace* seperti yang ditunjukkan gambar 3.2 diolah dengan menggunakan *awk script*, nilai dari setiap parameter hasil akan didapat. Dari keempat parameter hasil tersebut, nilainya akan dibandingkan mana dari varian TCP dan manajemen antrian yang memiliki kinerja terbaik untuk mengatasi *congestion* dalam jaringan.

3.6 Analisis Hasil

Pada penelitian ini, setelah mendapat hasil dari pengujian TCP Vegas dan TCP New Reno menggunakan antrian *Random Early Detection* dengan skema penambahan *min thresh* dan penambahan *max thresh* dan *Droptail* dengan skema penambahan kapasitas *buffer* pada NS-2, selanjutnya akan dilakukan analisis hasil serta perbandingan kinerja untuk melihat manakah varian TCP dan manajemen antrian yang memiliki kinerja terbaik dalam mengatasi *congestion* dengan melihat hasil dari empat parameter yang telah ditentukan yaitu *packet delivery ratio*, *throughput*, *delay* dan *packet drop*.

3.7 Kesimpulan Dan Saran

Tahapan terakhir merupakan penarikan kesimpulan dan pemberian saran. Dimana penarikan kesimpulan dilakukan untuk menjawab rumusan masalah yang telah ditentukan di bab satu. Sedangkan pemberian saran merupakan masukan yang diberikan oleh penulis yang dapat digunakan untuk memperbaiki atau mengembangkan penelitian selanjutnya.