

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil percobaan, pengambilan data serta analisis yang telah dilakukan pada bab sebelumnya, maka dapat disimpulkan bahwa :

1. Berdasarkan algoritma penjadwalan aktif yang telah diuji, dipilih satu algoritma penjadwalan aktif terefektif dan dinilai terbaik, yaitu algoritma penjadwalan *Shortest Expected Delay* (SED) yang menghasilkan *response time* sebesar 1.0068 detik dengan nilai *throughput* rata-rata sebesar 1.7508 kbps.
2. Berdasarkan algoritma penjadwalan pasif yang telah diuji, dipilih satu algoritma penjadwalan pasif terefektif dan dinilai terbaik, yaitu algoritma penjadwalan *Destination Hashing* (DH) yang menghasilkan *response time* sebesar 1.006 detik dan nilai *throughput* rata-rata sebesar 1.752 kbps.
3. Berdasarkan sepuluh algoritma penjadwalan yang telah diuji, dipilih satu algoritma penjadwalan terefektif dan dinilai terbaik, yaitu algoritma penjadwalan *Destination Hashing* (DH) yang menghasilkan *response time* sebesar 1,006 detik dan nilai *throughput* rata-rata sebesar 1.752 kbps.
4. Berdasarkan dari dugaan algoritma *Shortest Expected Delay* (SED) dan *Never Queue* (NQ) yang tidak berjalan sesuai alur algoritma, dugaan ini kurang tepat dan hal ini dipengaruhi oleh adanya *cache* sehingga waktu tunggu akan lebih cepat ke *real server* yang sudah terhubung sebelumnya, oleh karena itu trafik akan selalu masuk ke *real server* yang sama.

5.2 Saran

Berdasarkan dari simpulan dan temuan dari penelitian yang telah dilakukan, maka peneliti ingin memberikan beberapa saran untuk pengembangan pada penelitian selanjutnya, yaitu :

1. Jika ada penelitian lebih lanjut dengan menggunakan topik yang sama, dapat dicoba menggunakan fitur *keepalived* pada *load balancer* untuk mengetahui proses pembagian *request* saat ada pengurangan jumlah *DNS server* ditengah proses serta dapat menyimpan konfigurasi *load balancing* sehingga tidak akan hilang saat komputer di *reboot*.
2. Untuk pengembangan dan hasil yang lebih baik, implementasi *load balancing* trafik DNS dengan topologi *IP Tunneling*, lebih baik jumlah *client* dan *real server* yang terhubung untuk melakukan *request* ditambah sehingga alur algoritma menjadi lebih jelas perbedaannya.
3. Jika ada penelitian lebih lanjut dengan menggunakan topik yang sama, dapat dicoba menggunakan *load balancing* dengan metode *direct routing*. Secara teoritis, metode ini memiliki *response time* lebih rendah.

