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 paling efektif dan dinilai terbaik diterapkan pada FTP *server*, yaitu algoritma penjadwalan *Locality Based Least Connection* (LBLC) yang menghasilkan *response time* sebesar 0.6066 s dengan nilai *throughput* rata-rata sebesar 43 kb/s.
2. Berdasarkan algoritma penjadwalan pasif yang telah diuji, dipilih satu algoritma penjadwalan pasif paling efektif dan dinilai terbaik, yaitu algoritma penjadwalan *Round Robin* (RR) yang menghasilkan *response time* sebesar 0.6698 s dan nilai *throughput* rata-rata sebesar 42 kb/s.
3. Berdasarkan sepuluh algoritma penjadwalan yang telah diuji, dipilih satu algoritma penjadwalan paling efektif dan dinilai terbaik, yaitu algoritma penjadwalan *Locality Based Least Connection* (LBLC) yang menghasilkan *response time* sebesar 0.6066 s dan nilai *throughput* rata-rata sebesar 43 kb/s.

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. Penelitian lebih lanjut dengan menggunakan topik yang sama sebaiknya dapat dicoba menggunakan fitur *keepalives* pada *load balancer* untuk mengetahui proses pembagian *request* saat ada pengurangan jumlah FTP *server* di tengah 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 pada FTP *server* dengan topologi IP *Tunneling*, lebih baik jumlah *client* dan *real server* yang terhubung untuk melakukan *request* ditambah sehingga alur algoritma menjadi lebih jelas perbedaannya dan kapasitas masing-masing *server* dibeda-bedakan.
3. Penelitian lebih lanjut juga bisa menggunakan aplikasi selain FTP *server*, karena berbeda aplikasi, algoritma penjadwalan yang efisien digunakan bisa jadi berbeda.