

## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil pengujian dan pengambilan data serta analisis yang telah dilakukan pada bab sebelumnya, maka dapat disimpulkan bahwa:

1. Penggunaan tipe data *transfer* yang digunakan seperti *FTP server*, menjadi salah satu faktor penyebab terjadinya fluktuasi *bandwith*, karena kebutuhan akan parameter kinerja jaringan. Sehingga nilai *response time* yang akan di dapatkan tidak terlalu mempunyai pengaruh yang besar, dibandingkan pengaruhnya pada besar dan kecilnya nilai *throughput* karena parameter kinerja jaringan. Berdasarkan sepuluh algoritma penjadwalan *Linux Virtual Server* yang telah diuji, maka dipilih satu algoritma penjadwalan *Linux Virtual Server* yang terbaik dan efektif. Algoritma penjadwalan *Destination Hashing* (DH) menghasilkan nilai rata-rata *response time* sebesar 1,34 ms dan rata-rata *throughput* sebesar 226,3 bps.
2. Algoritma penjadwalan yang baik adalah algoritma penjadwalan dengan *response time* yang kecil dan menghasilkan *throughput* yang besar. Berdasarkan sifat dan kinerja Algoritma penjadwalan *Destination Hashing* (DH) pada parameter sistem jaringan maka algoritma penjadwalan *Destination Hashing* (DH) di tetapkan sebagai algoritma penjadwalan yang terbaik dan efektif diterapkan pada layanan *FTP server*.

### 5.2 Saran

Untuk memberikan hasil komparasi yang lebih baik:

1. Jika ada penelitian lebih lanjut yang bertopik sama, sebaiknya menggunakan Minimum Requirements Server:
  - Processor: Intel Quad Core i5-2410M CPU @1.60 GHz
  - Memory 8 GB RAM, DDR 3
  - OS (64 bit): Windows Server, 10, 8, 7 and Linux Ubuntu 14.04
  - Storage HDD 500 GB
  - Ethernet Card: *Realtek* PCIe FE Family Controller
  - LAN CARD (NIC): 2 unit support GigaByte

2. Jika ada penelitian lebih lanjut yang bertopik sama dan spesifikasi perangkat keras *server* yang digunakan pada *load balance* kurang tinggi, bisa dicoba menggunakan metode *load balance* yang kedua, yaitu metode *direct routing*. Secara teoritis, metode ini mampu memberi beban lebih rendah.
3. Untuk perkembangan dan hasil lebih baik, implementasi *load balancing* pada *FTP server* dengan topologi NAT, ada baiknya menggunakan banyak komputer sebagai penyedia layanan (*High Availability Cluster*). Di mana prinsip kerja *high availability cluster* ketika *server* utama gagal menyediakan layanan *service*, maka *server* yang lain yang sudah ter-*cluster* akan menggantikan tugas dari *server* utama secara otomatis sehingga meningkatkan ketersediaan data.

