

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi jaringan pada saat sekarang ini sudah tidak bisa dielakkan, termasuk perkembangan dalam akses komunikasi internet. Peningkatan masyarakat yang sudah mulai sadar internet dan pelaksanaan IoT (*internet of thing*) menjadi bagian penyebab peningkatan penggunaan akses internet. Berdasarkan data dari APJII (Asosiasi Penyedia Jasa Internet Indonesia) di tahun 2009, dari total populasi sebanyak 264 juta jiwa penduduk Indonesia, ada sebanyak 171,17 juta jiwa atau sekitar 64,8 persen yang sudah terhubung ke internet[1]. dan Pada saat mengakses internet pastilah *client* menuju ke sebuah situs tertentu, misalnya detik.com, atau mahasiswa yang menuju ke portal kampusnya, pasti akan mengetikkan nama portalnya di *browser*. Nama yang di tampilkan pada *address bar* pada *browser* tersebut adalah DNS (*Domain Name System*).

DNS digunakan sebagai cara untuk memasuki sebuah situs, dengan mengetikkan DNS pada *address bar* di *browser*, maka *client* akan bisa mengunjungi situs tersebut. Sebenarnya, situs tidak langsung dipanggil untuk ditampilkan pada jendela *browser*, tetapi browser memanggil *IP Address* yang terdaftar dibalik DNS tersebut. Karena sebenarnya, untuk mengakses sebuah situs, *client* harus memanggil *IP Address* dari situs tersebut, dan untuk menghafal deretan angka IP tersebut bukanlah hal yang mudah. Terlebih jika harus mengakses banyak situs yang *IP Address*-nya juga sudah pasti berbeda. Dengan adanya DNS, *client* tidak perlu menghafal dan mengetikkan *IP Address* pada *address bar* dan hanya perlu mengetikkan DNS dari situs yang di tuju. Sehingga proses dalam pencarian internet lebih mudah dan tidak perlu ribet menghafal angka-angka, cukup mengingat nama dari situs tersebut. Contohnya, untuk mengakses Google, *client* seharusnya mengetikkan ip google yaitu 172.217.31.255. namun, dengan DNS *client* tidak perlu mengetik apalagi mengingat angka tersebut, cukup mengetikkan google.com maka jendela Google langsung terbuka.

Dikarenakan peningkatan penggunaan internet dan akses ke situs semakin banyak, maka kerja pada DNS akan semakin berat, karena diakses pada saat yang bersamaan, untuk membuat kerja DNS menjadi lebih stabil, dibutuhkanlah sistem jaringan yang lebih baik, lebih cepat, agar *client* merasa puas. Salah satu caranya dengan menggunakan *load balancing*. *Load balancing* adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, dan menghindari *overload* pada salah satu jalur koneksi[2]. Sehingga, dengan adanya *load balancing*, kecepatan akses *client* semakin cepat. Yang melakukan kegiatan *load balancing* dinamakan *Load Balancer*. *Load balancer* ini tidak hanya berguna untuk mempercepat arus koneksi, tetapi juga dapat mengalihkan *client* ke server lain jika salah satu server rusak dengan isi data yang sama. Sehingga, *client* tidak menyadari bahwa pada server tujuannya sedang terjadi masalah.

Layanan *load balancing* memungkinkan pengaksesan sumber daya dalam jaringan didistribusikan ke beberapa *host*, agar tidak terpusat, sehingga kinerja jaringan komputer menjadi lebih stabil[3]. Oleh karena itu dibutuhkanlah algoritma untuk menjalankan tugasnya. Berbagai teknik dan algoritma pembagian beban bisa dilakukan oleh perangkat *load balancing* yang kompleks, yang bertujuan untuk menyesuaikan pembagian beban dengan karakteristik dari masing-masing pengguna internet, sehingga saat *client* mengakses sebuah situs tidak terjadi pembebanan berlebih kepada salah satu server, dan peluang terjadinya *server down* menjadi jauh berkurang.

Agar jaringan lancar, aman, dan tidak terganggu, dibutuhkan sebuah topologi jaringan yang mengatur bagaimana susunan komunikasi antara *client*, *load balancer*, dan server yang dituju. Topologi yang sering dikenal adalah Topologi Bus, selain topologi ini, masih ada topologi yang jarang disebutkan, yaitu topologi *Tunneling* yang juga merupakan salah satu teknik untuk *IP Load Balancing*. *Tunneling* merupakan salah satu cara untuk membangun sebuah jalur antar router di atas sebuah koneksi TCP/IP [4]. Dengan dihubungkannya *load balancer* dengan server tujuan menggunakan *tunneling*, menyebabkan IP *client* yang terhubung dengan jaringan tidak dikenal oleh server-server di internet, karena dengan *tunneling* akses jaringan yang

dilakukan oleh *client* menjadi *private* dan tidak akan bisa diganggu oleh pihak lain di internet. Oleh karena itu, keamanan *client* dari kejahatan *cyber* sangat tinggi karena komunikasi jaringan yang tidak bisa dideteksi oleh pihak lain. Dengan akses *client* terhadap internet yang meningkat, serta permasalahan yang sudah dijabarkan sebelumnya itulah, peneliti ingin membuat DNS yang memiliki *load balancer* dengan mengaplikasikan *tunneling*, dengan tujuan bisa meningkatkan keamanan dan kelancaran dalam melakukan koneksi data.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, masalah yang akan dibahas yaitu,

1. Bagaimana cara mengimplementasikan load balancing dengan variasi algoritma pada aplikasi DNS?
2. Bagaimana cara mengetahui kinerja dari load balancing dengan variasi algoritma pada aplikasi DNS?

1.3 Batasan Masalah

Agar dalam pengerjaan penelitian ini lebih terfokus, maka ditentukan batasan masalahnya, sebagai berikut,

1. *Server load balancing* dibuat dengan menggunakan LVS (*Linux Virtual Server*).
2. Topologi *load balancing* yang digunakan adalah *IP Tunneling*
3. Kinerja load balancing yang diperhatikan adalah *response time*, dan *throughput*.
4. Protokol jaringan yang dianalisis adalah protokol UDP.

1.4 Tujuan Penelitian

Penelitian ini dilaksanakan dengan tujuan sebagai berikut :

1. Mengetahui cara kerja sistem *load balancing* pada aplikasi DNS.

2. Mengetahui kinerja load balancing pada trafik DNS dengan variasi algoritma penjadwalan dan topologi *IP Tunneling*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah mendapatkan solusi dari permasalahan keamanan dalam komunikasi data lewat jaringan internet yang rawan, serta kecepatan akses meningkat.

1.6 Sistematika Penulisan

Sistematika penulisan merupakan langkah-langkah dalam membuat tugas akhir. Yang bertujuan untuk lebih memperjelas dan mempermudah dalam penyampaian informasi terkait pembahasan masalah, dengan susunan sebagai berikut:

1.6.1. BAB I : PENDAHULUAN

Berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan Tugas Akhir.

1.6.2. BAB II : TINJAUAN PUSTAKA

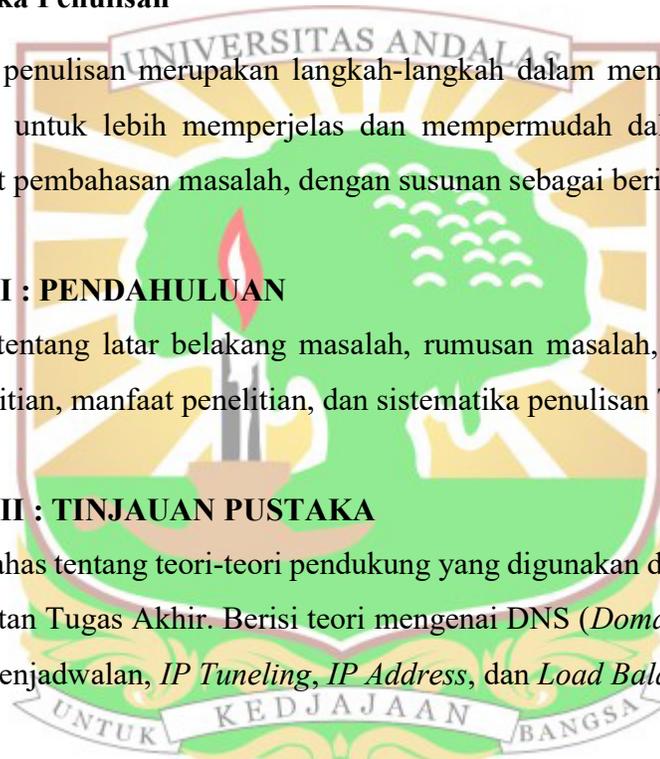
Membahas tentang teori-teori pendukung yang digunakan dalam perencanaan dan pembuatan Tugas Akhir. Berisi teori mengenai DNS (*Domain Name System*), Algoritma Penjadwalan, *IP Tuneling*, *IP Address*, dan *Load Balancing*.

1.6.3. BAB III : METODOLOGI PENELITIAN

Berisikan tentang perancangan penelitian, penjelasan tahap-tahap penelitian dan data yang akan diambil pada penelitian.

1.6.4. BAB IV : HASIL DAN PEMBAHASAN

Berisi tentang pembahasan aplikasi dan jalannya sistem, dilihat dari *response time* dan *throughput*-nya.



1.6.5. BAB VI : PENUTUP

Penutup berisi kesimpulan beserta saran untuk dijadikan bahan pertimbangan dalam pengembangan penelitian selanjutnya.

