

ABSTRAK

Video streaming merupakan suatu layanan yang memungkinkan sebuah *server* untuk mengirimkan *video* ke beberapa *user* yang berada pada suatu jaringan. Sebagai contoh layanan *video streaming* dapat dimanfaatkan untuk *long distance learning*. Untuk menyediakan layanan *long distance learning* dibutuhkan suatu jaringan yang dapat menjaga privasi *client* dan menyediakan jaminan QoS. Hal ini dapat diatasi dengan teknologi *tunneling* pada jaringan *Virtual Private Network* (VPN). Tetapi *tunneling* ini memiliki kelemahan karena kompleksitas jaringan dan mahalnya perangkat yang digunakan. Sehingga lahirlah teknologi *Virtual Private LAN Service* (VPLS) yang dapat mengatasi masalah tersebut. Dengan penambahan teknologi *multicast* pada VPLS diharapkan dapat meningkatkan QoS layanan yang bersifat *point-to-multipoint* seperti *video streaming*.

Pada tugas akhir ini dilakukan pengujian jaringan *Multicast VPLS* untuk layanan *video streaming*. Skenario pengujian yang dilakukan antara lain untuk melihat perbandingan performansi dari jaringan OSPF dan VPLS, dan pengaruh implementasi *multicast* pada VPLS, pengaruh variasi *bandwidth*, variasi *bitrate*, dan variasi jumlah *client* yang mengakses layanan *video streaming*.

Dari hasil pengujian dan analisis implementasi sistem *Multicast VPLS* didapatkan hasil yaitu jaringan VPLS memiliki QoS yang lebih baik daripada jaringan OSPF karena dapat mengurangi *delay* sampai 20.03%, meningkatkan *throughput* sampai 23.13%, dan mengurangi *packet loss* sampai 79.91%. Penambahan teknologi *multicast* terbukti dapat meningkatkan performansi jaringan VPLS yaitu dapat mengurangi *delay* sampai 25.66%, meningkatkan *throughput* 34.27%, tetapi *packet loss* yang dihasilkan oleh *multicast* lebih besar yaitu memiliki selisih sampai 3.54% dibandingkan *unicast*. Kemudian *bandwidth*, *bitrate*, dan juga jumlah *client* yang mengakses layanan *video streaming* terbukti mempengaruhi performansi dari jaringan *multicast VPLS* untuk layanan *video streaming*.

Kata kunci: QoS, MPLS, VPLS, *Multicast*, *Tunneling*