

ABSTRAK

Kebutuhan akan pelayanan yang cepat, nyaman dan murah dengan sadar telah menggerakkan riset di bidang teknologi informasi. Virtualisasi pun berkembang pesat. Dengan virtualisasi kita bisa terlepas dari ketergantungan fisik karena dengan virtualisasi, fisik yang berjumlah satu berfungsi sama dengan fisik berjumlah banyak. Dengan Virtual Local Area Network (VLAN) kita bisa memiliki beberapa jaringan dengan cukup satu manageable switch. Dan untuk menghubungkan antar VLAN (inter VLAN) cukup menggunakan 1 router 1 *interface*. Sehingga banyak biaya yang bisa dihemat. VLAN adalah pengelompokan logikal dari user dan sumber daya network yang terhubung ke port-port yang telah ditentukan secara administratif pada sebuah switch.

Karena layanan VoIP (Voice over IP) merupakan layanan yang realtime, maka VLAN yang telah diimplementasikan tersebut harus bersifat *high availability*, yaitu tingkat ketersediaan dan pelayanan harus 100% selalu ada. Pada tugas akhir ini akan diimplementasikan Virtual Redundancy Router Protocol (VRRP) untuk mewujudkan high availability inter VLAN tersebut. VRRP melakukan fungsi pengambilalihan tugas perutean terhadap paket-paket data yang masuk ke *virtual router group* ketika router master yang bertindak sebagai router utama mengalami *down* atau gangguan kinerja.

Dari hasil pengukuran dan analisis implementasi system InterVLAN VRRP didapatkan penurunan kualitas VoIP yaitu dikarenakan bertambahnya header sebesar 24 byte. Kondisi terburuk didapatkan dari interVLAN VRRP *master backup mode* namun QoS masih memenuhi standar. Hal itu dapat diperbaiki pada VRRP Load Sharing mode yang terbukti stabil dengan nilai QoS yang lebih baik. Pada scenario 2 (*testbad*), dapat dibuktikan bahwa VRRP mampu menjaga kualitas VoIP dengan terpenuhinya standar QoS meskipun terjadi gagal link. Pada scenario 3 juga didapatkan bahwa VRRP bukan merupakan solusi yang mahal karena terbukti bisa berfungsi sebagai *Load Sharing* yang mampu meningkatkan performansi jaringan.

Kata kunci : VRRP, VLAN, Inter VLAN, QoS, *Downtime*