

## ABSTRAK

WiMAX (*World Interoperability for Microwave Access*) merupakan teknologi yang diimplimentasikan oleh IEEE 802.16 yaitu evolusi dari teknologi *Broadband Wireless Access* (WBA) sebelumnya dengan penambahan fitur-fitur yang lebih menarik dan mampu menyediakan layanan jasa dengan akses berkecepatan tinggi (hingga 75 Mbps) dan didesain untuk kondisi *Non-LOS*(*non- Line of Sight*). Dalam perkembangannya, WiMAX nantinya tidak hanya diperuntukkan bagi market *fixed* saja tapi juga yang bersifat *portable* bahkan *mobile*. Dengan kecepatan akses yang tinggi inilah maka WiMAX layak di aplikasikan untuk “*last mile*” *broadband connections backhaul* dan *high speed enterprise*.

Karena begitu banyaknya aplikasi WiMAX baik untuk aplikasi yang *real time* maupun *non real time* inilah maka perlu dilakukan analisis performansi (QoS). Untuk itu perlu dilakukan suatu perbandingan algoritma *scheduling* untuk memilih mana yang lebih efektif digunakan guna meningkatkan QoS tersebut. Pada Tugas Akhir ini akan disimulasikan tiga skema penjadwalan dalam system antrian yaitu algoritma *scheduling Weighted Round Robin* (WRR), *Weighted Fair Queueing* (WFQ) dan *Modified Deficit Round Robin* (MDRR). Dalam Pengujian QoS ini dibutuhkan beberapa parameter yaitu *throughput*, *delay* dan *data drop* (*packet loss*). Dimana ini semua akan disimulasikan dengan menggunakan *software* OPNET Modeler 14.0

Dari beberapa skenario simulasi yang dilakukan dan dianalisa dapat disimpulkan bahwa WRR,WFQ dan MDRR merupakan algoritma yang mampu menunjukkan performansi yang baik dan memiliki keunggulan pada kondisi jaringan tertentu. WFQ bagus digunakan pada jaringan yang aplikasi didominasi oleh layanan rTPS yang bersifat delay intolerant. Sedangkan WRR dan MDRR bagus digunakan pada jaringan dengan jumlah user yang semakin meningkat dengan *mobilitas user* yang cukup tinggi.

**Kata Kunci** : WiMAX, algoritma *scheduling Weighted Round Robin* (WRR), *Weighted Fair Queueing* (WFQ) dan *Modified Deficit Round Robin* (MDRR), *throughput*, *delay* dan *datadrop*