

ABSTRAK

Dengan perkembangan teknologi dan media yang berbasis internet begitu cepat, tentunya penggunaan Internet akan semakin tinggi. Saat ini, standar komunikasi yang digunakan di seluruh dunia adalah *Transmission Control Protocol* atau *Internet Protocol Suite* (TCP/IP) yang arsitekturnya berpusat pada model komunikasi *host-to-host*. *Named Data Networking* dikembangkan untuk melakukan *cache* secara efisien dapat dilakukan dengan strategi *cache policy* dengan mengalokasikan *content store* secara homogen maupun heterogen. Dalam *cache* homogen, data paket yang tersimpan di router jaringan memiliki ukuran *content store* yang sama. Sementara itu, *cache* heterogen menyimpan data paket di setiap router dalam jaringan memiliki ukuran *content store* yang berbeda. Ada beberapa teknik optimasi berdasarkan *replacement algorithm* dan *placement algorithm*. Pada *replacement algorithm* diantaranya *Least Recently Used* (LRU) dan *Least Frequently Used* (LFU). Pada *placement algorithm* diantaranya *Leave Copy Everywhere* (LCE), *Leave Copy Down* (LCD), *Probability*, dan *Edge*.

Pada Tugas Akhir ini dibahas *cache policy* penempatan *content store* (CS) dengan ukuran yang homogen dan heterogen serta penggunaan jenis algoritma *cache replacement* dan *placement*. Performansi sistem akan dilihat dari parameter *cache hit ratio*, *delay*, dan *packet drop*.

Pada penelitian ini dilakukan simulasi untuk menguji skema *Cache placement* di jaringan NDN. Pengujian dilakukan dengan malakukan perubahan *content store*, perubahan ukuran frekuensi *interest* dan perubahan ukuran Zipf eksponensial. *Cache placement* LCE secara keseluruhan memiliki performansi terbaik di bandingkan tiga *Cache placement* lainnya. Untuk *Cache placement Probability*, memiliki performansi lebih baik mengungguli LCE saat skema perubahan frekuensi *interest* di frekuensi 200 dan 500. Hal ini membuktikan bahwa *Cache placement Probability* bisa memiliki performansi lebih baik saat jaringan yang sibuk.

Kata Kunci: *Named Data Networking, LRU, LFU, Caching Policy*