

Abstrak

Algoritma Lempel Ziv Welch (LZW) menggunakan teknik *dictionary* dalam melakukan kompresi, *dictionary* merupakan sebuah kamus referensi untuk menyimpan berbagai macam kombinasi karakter. Namun *dictionary* yang bersifat *array based* memiliki kelemahan dalam hal memasukkan *string* dan pencarian *string* secara *brute force* untuk itu maka diterapkan *Binary Search Tree* pada *dictionary LZW*. Kedua metode memiliki persamaan yaitu sama-sama melakukan proses inisialisasi 256 karakter *ASCII*. Sedangkan Algoritma Huffman menggunakan teknik statistik dalam melakukan kompresi. Fase pertama yang dilakukan adalah menghitung frekuensi kemunculan simbol kemudian membuat sebuah pohon biner yang berisikan urutan kemunculan simbol dari yang terbanyak sampai yang paling sedikit. Kode Huffman dari setiap simbol adalah *output* kompresi berupa rangkaian biner yang dibaca dari daun sampai akar yang bersangkutan.

Pada tugas akhir ini akan menganalisis kinerja dua algoritma kompresi yaitu Huffman dan LZW *Binary Search Tree* dalam proses kompresi maupun dekompresi sebuah file *plaintext* dan *ciphertext* Rijndael, dengan mengamati parameter rasio kompresi dan waktu kompresi dan dekompresi yang dibutuhkan.

Berdasarkan hasil penelitian yang dilakukan maka dapat disimpulkan bahwa penerapan struktur *dictionary* berupa *binary search tree* pada algoritma LZW masih belum optimal jika dibandingkan terhadap algoritma Huffman.

Kata Kunci : Kompresi, LZW, Binary Search Tree, Dictionary, Huffman, rasio, waktu