

## ABSTRAK

Pemampatan data adalah proses mengubah file data masukan (data asli) menjadi file data keluaran supaya ukuran file data menjadi lebih kecil. Prinsip kerja dari pemampatan yaitu dengan mengurangi redundansi dalam file data masukan. Terdapat banyak algoritma pemampatan yang mempunyai prinsip kerja berbeda-beda dan menghasilkan data keluaran yang ukurannya berbeda pula, diantaranya adalah algoritma pemampatan *Run Length*, *Half-byte*, dan *Huffman*. Untuk mengetahui algoritma mana yang terbaik, dilakukan perbandingan untuk ketiga algoritma pemampatan tersebut.

Perbandingan hanya dilakukan untuk berkas teks (\*.txt dan \*.doc) dengan menganalisa algoritma dan menguji program untuk beberapa file. Kriteria perbandingan meliputi kompleksitas waktu (waktu tempuh) yang dicapai, rasio pemampatan, dan ukuran file hasil pemampatan dari ketiga algoritma.

Hasil analisis algoritma untuk ketiga algoritma pemampatan menunjukkan bahwa untuk waktu tempuh algoritma *Run Length* dan *Half-Byte* masing-masing adalah  $O(n)$ , sedangkan waktu tempuh terbesar dimiliki oleh algoritma *Huffman*, yaitu  $O(n^4)$ . Hal ini sesuai dengan rata-rata waktu yang dibutuhkan untuk proses pemampatan ketiga algoritma, yaitu untuk berkas file berekstensi \*.txt rata-rata waktu tercepat diperoleh dari algoritma *Half-Byte* dan untuk berkas file berekstensi \*.doc rata-rata waktu tercepat diperoleh dari algoritma *Run Length*, sedangkan pada algoritma *Huffman* dibutuhkan waktu yang lama untuk proses pemampatan untuk berkas file berekstensi \*.txt dan \*.doc.

Hasil rasio pemampatan terbaik dan ukuran berkas file hasil pemampatan yang terkecil diperoleh dari algoritma pemampatan *Huffman*.

## **ABSTRACT**

*Data compression is the process of converting an input data stream (the source stream or original data) into another data stream (the output, or the compressed stream) that has smaller size. The principle of data compression is by removing redundancy from the original data in the source files. There are many methods for data compression, which are based on different ideas, are suitable for different types of data and produce different results. Some of them are Run Length, Half-Byte, and Huffman. To know which algorithm is the best, a comparison of these algorithms needs to be performed.*

*The comparison was done on text files (\*.txt and \*.doc) by analyzing the algorithms and testing the programs for some files. The criteria that are used are time complexity (running time), compression ratio, and the size of the compressed file of the three algorithms.*

*The result of the algorithm analysis reveals that the running time of Run Length and Half-Byte algorithm respectively is  $O(n)$ , while the running time of Huffman algorithm is  $O(n^4)$ . It goes along with the average time for the process, that is for files with \*.txt extension, the fastest time is Half-Byte algorithm and for files with \*.doc extension the fastest time is Run Length algorithm, while Huffman algorithm takes long time to compress both types of file. Huffman algorithm produces the best result in term of compression ratio and produces the smallest size of the compressed files.*