

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan sehari-hari sering dijumpai kegiatan-kegiatan yang membutuhkan adanya proses pengurutan. Saat menggunakan buku telepon untuk mencari nomor telepon dari satu nama, akan ditemukan beribu-ribu data yang terdiri dari nama pelanggan, alamat, dan nomor teleponnya. Jika nama-nama yang ada tidak diurutkan berdasarkan abjad maka terdapat kesulitan untuk menemukannya. Demikian juga saat menggunakan kamus untuk mencari suatu kata, akan dibutuhkan waktu yang cukup lama jika kata-kata yang terdapat dalam kamus tersebut tidak terurutkan. Hal ini membuktikan bahwa pengurutan merupakan suatu proses yang sangat penting terutama dalam pencarian data.

Pengurutan (*sorting*) merupakan suatu proses menyusun objek-objek yang ada dengan menggunakan aturan-aturan tertentu ke dalam suatu urutan. Tujuan *sorting* adalah untuk mempermudah dalam proses pencarian suatu data. Dengan demikian berarti *sorting* sangat erat hubungannya dengan pengolahan data, bahkan merupakan bagian yang penting dalam pengolahan data. *Sorting* merupakan subjek yang dapat menunjukkan bermacam-macam algoritma yang mempunyai satu tujuan. Terdapat beberapa algoritma untuk *sorting* dan salah satu diantaranya adalah *heapsort*.

1.2 Permasalahan

Algoritma heapsort merupakan algoritma yang digunakan untuk pengurutan atau sorting. Dalam tugas akhir ini akan dibahas :

1. algoritma heapsort untuk pengurutan
2. analisa algoritma heapsort terutama dilihat dari running time-nya
3. implementasi algoritma heapsort menggunakan bahasa pemrograman Pascal, khususnya running time-nya

1.3 Pembatasan Masalah

Pengurutan yang akan dilakukan adalah pengurutan elemen data dalam suatu larik secara ascending . Pada algoritma heapsort , larik yang terdiri atas kunci disajikan sebagai suatu pohon biner dengan sifat-sifat khusus yang disebut *heap*, dimana kunci suatu simpul tidak boleh lebih kecil daripada kunci simpul anak. Dengan menukar kunci simpul terbesar dengan kunci simpul terakhir , akan diperoleh elemen terakhir dari larik yang akan diurutkan. Selanjutnya dengan mengulang kedua proses tersebut (membentuk heap dan menukar kunci) , akan diperoleh larik dengan kunci yang terurut. Adapun implementasi algoritma menggunakan bahasa pemrograman Pascal hanya ditunjukkan pada pengurutan angka untuk diperoleh running time-nya.

1.4 Sistematika Penulisan

Bab I berisi tentang latar belakang, perumusan masalah, pembatasan masalah serta sistematika penulisan .

Bab II berisi tentang konsep dasar graf dan pohon, dasar-dasar algoritma, kompleksitas algoritma, pengurutan, running time program, serta bahasa pemrograman Pascal.

Bab III berisi tentang pembahasan algoritma heapsort beserta contoh dan analisisnya, khususnya kompleksitas waktu.

Bab IV berisi tentang penjelasan program grafik running time heapsort menggunakan bahasa pemrograman Pascal.

Bab V merupakan kesimpulan dari tugas akhir ini.

