

**PERBANDINGAN BEBERAPA METODE PENGURUTAN DATA PADA
TURBO PASCAL DITINJAU DARI SEGI KECEPATAN**

Haeni Budiati

Ilmu Komputer, FMIPA, UKRIM

Abstraksi

Pengurutan data merupakan suatu proses penyusunan urutan objek dengan aturan tertentu. Proses pengurutan data dilakukan dengan mengurutkan data random (acak) menjadi sebuah himpunan data tersusun dan terorganisir.

Kata Kunci : Randomize

1. PENDAHULUAN

Dewasa ini ilmu komputer telah berkembang dengan pesatnya yang disertai juga dengan perkembangan di bidang hardware (perangkat keras), dan software (perangkat lunak). Masing-masing bidang saling bersaing untuk mencapai yang terbaik dan cenderung untuk menunjukkan ke konsumen (user).

Secara umum pengurutan data bisa didefinisikan sebagai suatu proses untuk menyusun kembali himpunan objek dengan menggunakan aturan tertentu. Ada dua jenis pengurutan data, yaitu pengurutan secara urut naik (*ascending*), yaitu dari data yang nilainya paling kecil sampai data yang nilainya paling besar, dan pengurutan data secara urut turun (*descending*) yaitu data yang mempunyai nilai paling besar sampai paling kecil.

1.1. RUMUSAN MASALAH

Mempertimbangkan memudahkan penyelesaian masalah diatas, penulis akan menganalisa waktu yang dipakai dari masing-masing metode sorting sebagai faktor uji kecepatan proses.

1.2. TUJUAN

Adapun tujuan dari penulisan penelitian ini adalah :

1. Agar dapat lebih mengetahui metode mana yang mempunyai kecepatan yang baik dalam mengurutkan data.
2. Agar lebih memahami kelebihan dan kekurangan masing-masing metode pengurutan dalam hal kompleksitas algoritma dan banyaknya iterasi.

1.3. LANDASAN TEORI

Pencarian suatu obyek dalam himpunan yang terurut jauh lebih cepat dibandingkan pada himpunan yang tidak terurut (acak). Pada bahasa pemrograman Pascal menyediakan paling sedikit 2 fungsi dan prosedur untuk membentuk bilangan acak, masing-masing adalah *prosedur Randomize* dan *fungsi Random*.

Pengurutan Data adalah Suatu proses penyelesaian pada data acak menjadi data tidak acak (terurutkan). Setiap pemrosesan data yang masih manual biasanya sangat lambat sehingga membutuhkan waktu yang cukup lama. Secara umum pengurutan data bisa didefinisikan sebagai proses untuk menyusun kembali himpunan obyek menggunakan aturan-aturan tertentu. Pengurutan data juga merupakan salah satu cara mengorganisir data didalam meningkatkan efisiensi untuk menyelesaikan suatu proses permasalahan..

Metode-metode pengurutan diantaranya sebagai berikut :

1.3.1. Metode Radixsort

Metode Radixsort adalah metode pengurutan yang didasarkan pada nilai sesungguhnya dari suatu bilangan yang akan diurutkan. Dalam prosesnya metode ini juga memanfaatkan Disk (*Most Significant Digit*) yaitu setiap posisi digit paling kiri disebut signifikan terbesar, sedangkan DSB (*Least Significant Digit*) yaitu digit yang posisinya berada paling kanan disebut signifikan terkecil. Proses iterasi akan lebih mudah dikerjakan bila setiap bilangan ditemukan cacah digit yang sama. Setiap langkah pada proses pengerjaan akan berawal dari proses pemindahan data kelompok

berdasarkan posisi digit satuan, puluhan, ratusan, ribuan dan seterusnya tergantung dari banyaknya digit yang akan diolah, kemudian langkah selanjutnya melakukan pembagian kelompok yang dibandingkan berdasarkan digit signifikan terbesar (*Most Significant Digit*) yang sering disingkat dengan DSB dan digit signifikan terkecil (*Least Significant Digit*) disingkat dengan DSK. Proses ini diulang sampai dengan kondisi data sudah terurutkan.

1.3.2. Metode QuickSort

Metode Quicksort adalah metode pengurutan yang menempatkan suatu elemen pada posisinya yang tepat, kemudian mengurutkan subvektor kiri dan terakhir mengurutkan subvektor kanan. Dalam mengurutkan subvektor kiri atau kanan, ketiga langkah ini diulang kembali dengan menentukan batas-batas subvektor yang akan diproses. Batas akhir proses ditentukan apabila batas kiri subvektor sudah sama dengan atau lebih besar dari batas kanan. Untuk mengimplementasikan metode QuickSort digunakan 2 cara yaitu dengan Cara Rekursif dan cara Non rekursif.

1.4. Metode Seleksi (Selection Sort)

Metode seleksi adalah metode pengurutan yang mempunyai tujuan untuk menghasilkan sekumpulan data menjadi terurut. Proses pengerjaannya dengan mencari bagian dari elemen terkecil, selanjutnya melakukan perbandingan, kemudian proses penukaran dari posisi elemen pertama ke elemen lainnya dan proses pergeseran data sehingga menghasilkan data yang terurut. Kecepatan proses dipengaruhi oleh banyaknya data yang akan diurutkan, pembuatan algoritma. Untuk menentukan elemen terkecil yang akan kita tempatkan pada tujuan terlebih dahulu yang perlu kita perhatikan adalah semua elemen dalam larik sumber sehingga disebut *Multiple Source-one Destination*.

1.5. Metode Gelembung (BubbleSort)

Metode gelembung adalah metode yang mempunyai kegiatan prosesnya dapat menghasilkan kelompok data menjadi terurut. Cara kerjanya adalah dengan menentukan posisi elemen terbesar dan posisi elemen terkecil, pergeseran elemen, perbandingan antar elemen, penukaran dua buah elemen, metode ini juga dikatakan tidaklah efisien, setiap akan mengurutkan sebanyak data elemen yang iterasinya

kurang dari banyak data elemen tersebut, maka iterasinya harus dilaksanakan sampai banyaknya data elemen yang ada.

3.4. Metode ShellSort

Metode Shellsort adalah metode yang menempatkan suatu elemen dengan membandingkan elemen pertama dengan elemen yang lainnya pada jarak tertentu dari elemen pertama. Metode ini proses pengerjaannya dengan memanfaatkan minimal sepasang elemen, penukaran antar pasangan elemen, membentuk urutan dari masing-masing pasangan dengan membandingkan jarak antar elemen, mengkombinasikan dari masing-masing pasangan sehingga membentuk urutan baru untuk keseluruhan pasangan. Tujuan dari metode ini adalah untuk mempertinggi efektivitasnya.

3.5. Metode TreeSort

Metode TreeSort adalah Metode yang proses pengerjaannya untuk mengurutkan data dimana dasar proses pengerjaannya pada struktur data tak linier atau dinamakan dengan pohon tree. Struktur pohon ini menggambarkan hubungan yang bersifat hirarki antar elemen-elemen yang ada. Metode ini menjelaskan cara menyusun sebuah pohon didalam memperoleh informasi data yang sudah terurutkan dan tersimpan didalam pohon. Dalam membentuk pohon perlu diperhatikan kapan suatu simpul akan dipakai sebagai cabang kiri dan kapan sebagai cabang kanan. Salah satu aturan adalah bahwa informasi yang tersimpan pada cabang kiri nilainya selalu lebih kecil dari simpul akar dan informasi yang tersimpan dalam simpul akar nilainya selalu lebih kecil dari yang tersimpan pada cabang kanan.

3.6. Metode Penyisipan (InsertionSort)

Metode InsertionSort adalah Metode pengurutan dengan cara menyisipkan elemen lari pada posisi yang tepat. Pencarian posisi yang tepat dilakukan dengan pencarian beruntun dalam bentuk larik/pointer. Metode ini mempunyai tujuan seperti metode pengurutan lainnya, yaitu untuk mengurutkan data sehingga mempermudah didalam pencarian data. Untuk memperoleh proses, maka pengerjaannya dimulai dari perbandingan elemen pertama, kemudian proses pergeseran, pada proses pergeseran ini secara otomatis terjadi juga proses penyisipan.

3.7. Metode HeapSort

Metode HeapSort adalah metode yang kegiatannya untuk melakukan pemindahan data. Suatu heap adalah suatu array yang terdiri dari h_1, h_2, \dots, h_n . Sehingga dipenuhi persyaratan $h_i \leq h_{2i}$, $h_i \leq h_{2i+1}$. Bila suatu heap sudah berhasil dibentuk, dengan mudah diperoleh data dengan nilai key yang terkecil yaitu n . Tujuan dari metode ini memperbaiki pada penyusunan data yang sebaiknya dilakukan pada perbandingan nilai data pada daun atau leaves yang sudah terpilih tidak perlu dilakukan. Proses pertama kali yaitu dengan membentuk suatu heap tree dalam bentuk array kemudian melakukan pemilihan data terkecil yang terdapat pada akar untuk mendapatkan data yang terurut.

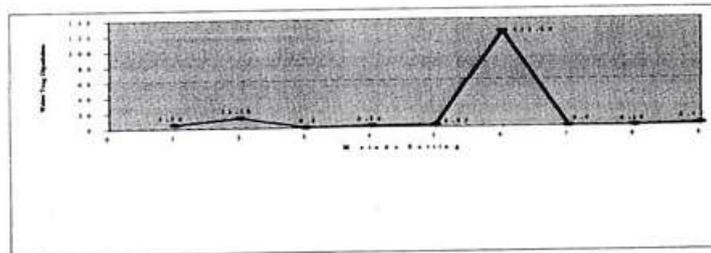
3.8. Metode MergeSort.

Metode MergeSort adalah metode dimana proses pengerjaannya dengan menggabungkan dua kelompok data menjadi satu kelompok data yang sudah terurutkan. Metode ini proses pengerjaannya menggunakan dua cara. Cara pertama adalah bila kita menemukan permasalahan dengan dua kelompok data yang diurutkan menjadi satu kelompok data, maka proses penyelesaiannya yaitu pertama kali melakukan perbandingan antara masing-masing kelompok data, kemudian dilakukan perbandingan antara masing-masing kelompok. Hasil dari perbandingan dimasukkan pada kelompok data gabungan. Proses ini dilakukan berulang-ulang sampai dua kelompok data tersebut menjadi urut dan membentuk satu kelompok data baru.

3. HASIL DAN PEMBAHASAN

Tabel I. Waktu Proses Untuk Data = 100
Dengan Menggunakan Komputer P 120

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|-----------|---------------------|
| 1. QuickSort | 5 detik | 32 perseratus detik |
| 2. InsertionSort | 13 detik | 12 perseratus detik |
| 3. ShellSort | 0 detik | 5 perseratus detik |
| 4. BubbleSort | 2 detik | 30 perseratus detik |
| 5. TreeSort | 2 detik | 52 perseratus detik |
| 6. SelectionSort | 123 detik | 58 perseratus detik |
| 7. HeapSort | 0 detik | 6 perseratus detik |
| 8. RadixSort | 0 detik | 10 perseratus detik |
| 9. MergeSort | 2 detik | 47 perseratus detik |



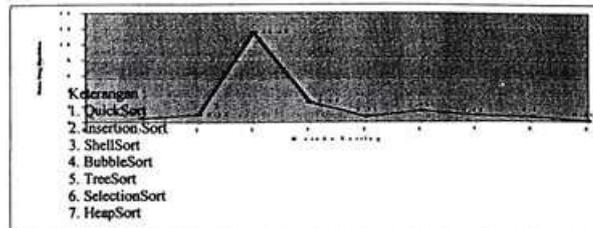
Hasil pengamatan pada tabel dan grafik diatas, dengan data sebanyak 100 data adalah :

1. Metode yang prosesnya membutuhkan waktu paling banyak adalah metode SelectionSort dengan waktu prosesnya 123 detik 58 perseratus detik.
2. Metode yang prosesnya membutuhkan waktu paling sedikit adalah metode ShellSort dengan waktu prosesnya 5 perseratus detik.

Tabel II. Waktu proses untuk data 3000
Dengan Menggunakan Komputer P 120

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|----------|---------------------|
| 1. QuickSort | 0 detik | 5 perseratus detik |
| 2. InsertionSort | 0 detik | 99 perseratus detik |
| 3. ShellSort | 11 detik | 48 perseratus detik |
| 4. BubbleSort | 2 detik | 53 perseratus detik |
| 5. TreeSort | 0 detik | 71 perseratus detik |
| 6. SelectionSort | 1 detik | 43 perseratus detik |
| 7. HeapSort | 0 detik | 84 perseratus detik |
| 8. RadixSort | 0 detik | 6 perseratus detik |
| 9. MergeSort | 0 detik | 10 perseratus detik |

Perbandingan Beberapa Metode Pengurutan Data Pada Turbo Pascal Ditinjau Dari Segi Kecepatan



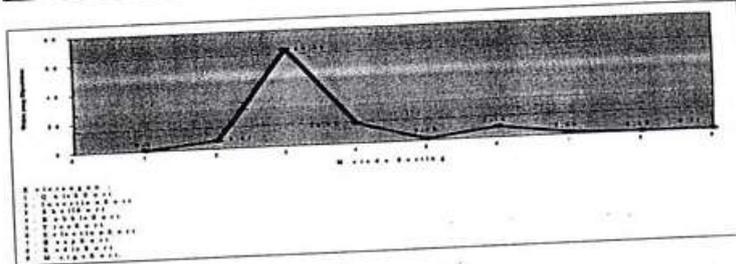
Hasil pengamatan pada tabel dan grafik diatas, dengan data sebanyak 100 data adalah :

1. Metode yang prosesnya membutuhkan waktu paling banyak adalah metode SelectionSort dengan waktu prosesnya 123 detik 58 perseratus detik.
2. Metode yang prosesnya membutuhkan waktu paling sedikit adalah metode ShellSort dengan waktu prosesnya 5 perseratus detik.

Tabel III. Waktu proses untuk data 7000
 Dengan Menggunakan Komputer P 120

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|----------|---------------------|
| 1. QuickSort | 0 detik | 5 perseratus detik |
| 2. InsertionSort | 3 detik | 84 perseratus detik |
| 3. ShellSort | 66 detik | 68 perseratus detik |
| 4. BubbleSort | 13 detik | 46 perseratus detik |
| 5. TreeSort | 0 detik | 82 perseratus detik |
| 6. SelectionSort | 7 detik | 14 perseratus detik |
| 7. HeapSort | 1 detik | 39 perseratus detik |

| | | |
|--------------|---------|---------------------|
| 8. RadixSort | 0 detik | 22 perseratus detik |
| 9. MergeSort | 0 detik | 11 perseratus detik |



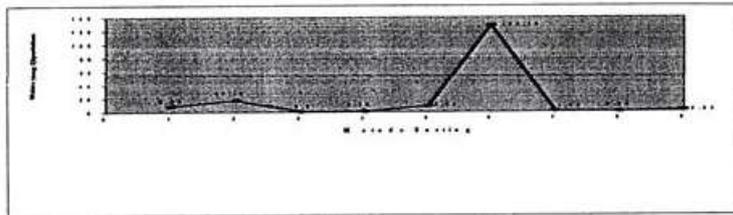
Hasil pengamatan pada tabel III dan grafik 3., dengan data sebanyak 7000 antara lain :

1. Metode yang waktu prosesnya paling besar ada 1 metode yaitu metode ShellSort, dengan waktu prosesnya adalah 66 detik 68 perseratus detik.
2. Metode yang waktu prosesnya paling sedikit ada 1 metode yaitu metode Quick Sort dengan waktu prosesnya adalah 5 perseratus detik.

Tabel IV. Waktu proses untuk data 100
Dengan Menggunakan Komputer 486 dx

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|----------|---------------------|
| 1. QuickSort | 8 detik | 45 perseratus detik |
| 2. InsertionSort | 17 detik | 10 perseratus detik |
| 3. ShellSort | 0 detik | 6 perseratus detik |
| 4. BubbleSort | 0 detik | 74 perseratus detik |
| 5. TreeSort | 8 detik | 61 perseratus detik |

| | | |
|------------------|-----------|---------------------|
| 6. SelectionSort | 128 detik | 20 perseratus detik |
| 7. HeapSort | 1 detik | 30 perseratus detik |
| 8. RadixSort | 0 detik | 28 perseratus detik |
| 9. MergeSort | 2 detik | 58 perseratus detik |



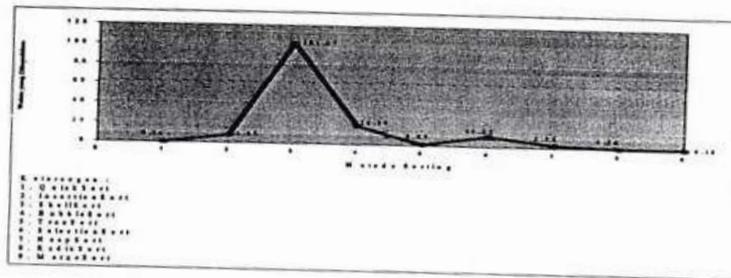
Hasil pengamatan pada tabel IV dan grafik 4, dengan data sebanyak 100 antara lain :

1. Metode yang waktu prosesnya paling besar ada 1 metode yaitu metode SelectionSort, dengan waktu prosesnya adalah 128 detik 20 perseratus detik.
2. Metode yang waktu prosesnya paling sedikit ada 1 metode yaitu metode Shell Sort dengan waktu prosesnya adalah 6 perseratus detik.

Tabel V. Waktu proses untuk data 3000
Dengan Menggunakan Komputer 486 dx

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|---------|---------------------|
| 1. QuickSort | 0 detik | 11 perseratus detik |
| 2. InsertionSort | 7 detik | 47 perseratus detik |

| | | |
|------------------|-----------|---------------------|
| 3. ShellSort | 101 detik | 67 perseratus detik |
| 4. BubbleSort | 18 detik | 30 perseratus detik |
| 5. TreeSort | 0 detik | 88 perseratus detik |
| 6. SelectionSort | 10 detik | 33 perseratus detik |
| 7. HeapSort | 2 detik | 35 perseratus detik |
| 8. RadixSort | 0 detik | 28 perseratus detik |
| 9. MergeSort | 0 detik | 18 perseratus detik |

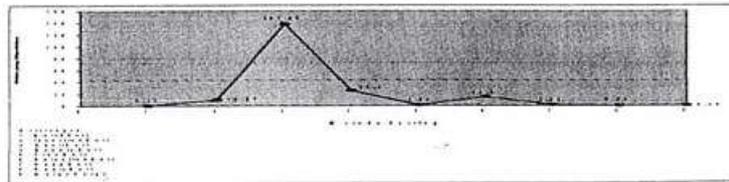


Hasil pengamatan pada tabel V dan grafik 5, dengan data sebanyak 6000 antara lain :

1. Metode yang waktu prosesnya paling besar ada 1 metode yaitu metode ShellSort, dengan waktu prosesnya adalah 101 detik 67 perseratus detik.
2. Metode yang waktu prosesnya paling sedikit ada 1 metode yaitu metode Quick Sort dengan waktu prosesnya adalah 11 perseratus detik.

Tabel VI. Waktu proses untuk data 7000
Dengan Menggunakan Komputer 486 dx

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|-----------|---------------------|
| 1. QuickSort | 0 detik | 13 perseratus detik |
| 2. InsertionSort | 10 detik | 27 perseratus detik |
| 3. ShellSort | 137 detik | 69 perseratus detik |
| 4. BubbleSort | 25 detik | 4 perseratus detik |
| 5. TreeSort | 1 detik | 26 perseratus detik |
| 6. SelectionSort | 14 detik | 6 perseratus detik |
| 7. HeapSort | 1 detik | 97 perseratus detik |
| 8. RadixSort | 0 detik | 33 perseratus detik |
| 9. MergeSort | 0 detik | 20 perseratus detik |



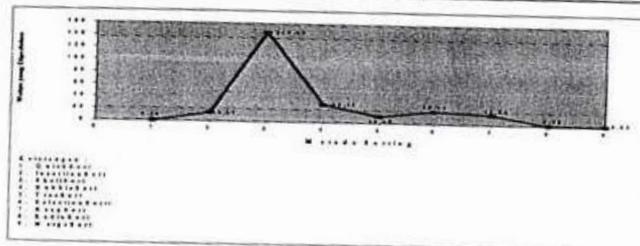
Hasil pengamatan pada tabel VI dan grafik 6, dengan data sebanyak 7000 antara lain :

1. Metode yang waktu prosesnya paling besar ada 1 metode yaitu metode ShellSort, dengan waktu prosesnya adalah 137 detik 69 perseratus detik.

2. Metode yang waktu prosesnya paling sedikit ada 1 metode yaitu metode Quick Sort dengan waktu prosesnya adalah 13 perseratus detik.

Tabel VII. Waktu proses untuk data 3000
Dengan Menggunakan Komputer 386 AT

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|-----------|---------------------|
| 1. QuickSort | 0 detik | 18 perseratus detik |
| 2. InsertionSort | 14 detik | 34 perseratus detik |
| 3. ShellSort | 145 detik | 49 perseratus detik |
| 4. BubbleSort | 29 detik | 16 perseratus detik |
| 5. TreeSort | 10 detik | 88 perseratus detik |
| 6. SelectionSort | 20 detik | 43 perseratus detik |
| 7. HeapSort | 15 detik | 95 perseratus detik |
| 8. RadixSort | 0 detik | 40 perseratus detik |
| 9. MergeSort | 0 detik | 22 perseratus detik |

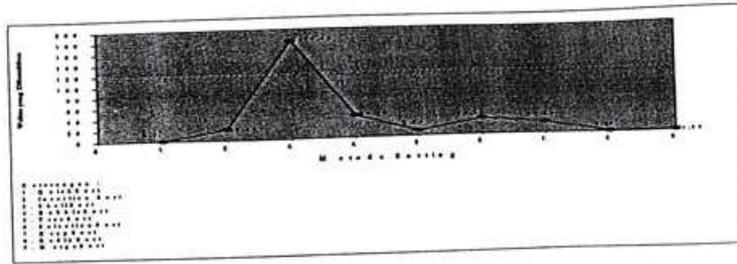


Hasil pengamatan pada tabel VIII dan grafik 8, dengan data sebanyak 3000 antara lain :

1. Metode yang waktu prosesnya paling besar ada 1 metode yaitu metode ShellSort, dengan waktu prosesnya adalah 145 detik 49 perseratus detik.
2. Metode yang waktu prosesnya paling sedikit ada 1 metode yaitu metode Quick Sort dengan waktu prosesnya adalah 18 perseratus detik.

Tabel VIII. Waktu proses untuk data 7000
Dengan Menggunakan Komputer 386 AT

| METODE SORT 2 WAKTU YANG DIPERLUKAN UNTUK SORTING | | |
|---|-----------|---------------------|
| 1. QuickSort | 0 detik | 20 perseratus detik |
| 2. InsertionSort | 19 detik | 99 perseratus detik |
| 3. ShellSort | 179 detik | 49 perseratus detik |
| 4. BubbleSort | 39 detik | 71 perseratus detik |
| 5. TreeSort | 8 detik | 46 perseratus detik |
| 6. SelectionSort | 28 detik | 76 perseratus detik |
| 7. HeapSort | 20 detik | 29 perseratus detik |
| 8. RadixSort | 1 detik | 11 perseratus detik |
| 9. MergeSort | 1 detik | 22 perseratus detik |



Hasil pengamatan pada tabel IX dan grafik 9, dengan data sebanyak 7000 antara lain :

1. Metode yang waktu prosesnya paling besar ada 1 metode yaitu metode ShellSort, dengan waktu prosesnya adalah 179 detik 49 perseratus detik.
2. Metode yang waktu prosesnya paling sedikit ada 1 metode yaitu metode Quick Sort dengan waktu prosesnya adalah 20 perseratus detik.

4, PEMBAHASAN

1. Menghitung waktu kecepatan.

Sebelum data diurutkan dari bilangan random menjadi bilangan terurut maka waktu terlebih dahulu direset 0 (nol). Kemudian saat data mulai diurutkan maka waktu dihitung mulai dari *starttimer*. Setelah data selesai diurutkan waktu diakhiri dengan *stoptimer*. Selisih waktu yang didapat adalah $starttimer - stoptimer$. Dari selisih waktu ini diketahui waktu yang diperlukan oleh setiap algoritma untuk menghitung waktu kecepatan pengurutan data .

2. Kompleksitas/Kerumitan Algoritma dan Jumlah Iterasi Algoritma

Untuk menentukan kompleksitas atau kerumitan suatu algoritma dapat ditinjau dari beberapa notasi O dimana :

| | |
|--------------|---|
| Notasi O | Prestasi dengan n tertentu |
| $O(1)$ | Konstanta (ukuran item tidak berpengaruh) |
| $O(\lg^n)^2$ | logaritma |
| $O(n)$ | linear (berbanding lurus) |

| | |
|--------------|--|
| $O(n \lg n)$ | $n \log 2 n$ (untuk algoritma bagi dua dan kerjakan seperti quicksort binary search) |
| $O(n^2)$ | Algoritma kuadrat |
| $O(n^3)$ | Cubik atau pangkat tiga |
| $O(mn)^3$ | Exponensial |
| $O(n!)$ | Faktorial |

3. Perbandingan antara setiap algoritma

Dari ke sembilan algoritma diatas maka terdapat perbandingan bahwa :

1. Algoritma RadixSort mempunyai kecepatan menghitung waktu secara umum dengan jumlah 40000 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang sederhana karena terdapat 230 buah
2. Algoritma QuickSort mempunyai kecepatan menghitung waktu secara khusus dengan jumlah 17039 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang sederhana karena terdapat 230 buah.
3. Algoritma Selection Sort mempunyai kecepatan menghitung waktu secara umum dengan jumlah 12497499 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang kompleks karena terdapat 1000 buah.
4. Algoritma BubbleSort mempunyai kecepatan menghitung waktu secara umum dengan jumlah 12490360 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang kompleks/rumit karena terdapat 1000 buah.
5. Algoritma ShellSort mempunyai kecepatan menghitung waktu secara umum dengan jumlah 24527762 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang kompleks/rumit karena terdapat 1000 buah.
6. Algoritma TreeSort mempunyai kecepatan menghitung waktu secara khusus dengan jumlah 76271 kali iterasi dari awal perhitungan sampai akhir perhitungan

waktu data yang diurutkan, dengan tingkat kompleksitas yang sederhana karena terdapat 100 buah.

7. Algoritma InsertionSort mempunyai kecepatan menghitung waktu secara umum dengan jumlah 6307272 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang kompleks karena terdapat 1000 buah.
8. Algoritma HeapSort mempunyai kecepatan menghitung waktu secara khusus dengan jumlah 61340 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang sederhana karena terdapat 230 buah.
9. Algoritma MergeSort mempunyai kecepatan menghitung waktu secara khusus dengan jumlah 55197 kali iterasi dari awal perhitungan sampai akhir perhitungan waktu data yang diurutkan, dengan tingkat kompleksitas yang sangat sederhana karena terdapat 230 buah.

4. Kesimpulan

Dari ke 9 (sembilan) metode pengurutan data yang telah dikemukakan sebelumnya maka dapat disimpulkan sebagai berikut :

1. Untuk jumlah data yang kecil yaitu data = 100, Metode algoritma ShellSort merupakan metode pengurutan data yang paling cepat sedangkan metode algoritma SelectionSort merupakan metode pengurutan data yang paling lama.
2. Untuk jumlah data yang besar yaitu data = 3000-7000, Metode algoritma QuickSort adalah merupakan metode pengurutan data yang paling cepat sedangkan metode ShellSort adalah merupakan metode pengurutan data yang paling lama.
3. Algoritma RadixSort, QuickSort, TreeSort, HeapSort, MergeSort adalah merupakan algoritma pengurutan data yang sederhana, sedangkan algoritma SelectionSort, BubbleSort, InsertionSort, ShellSort, dan TreeSort, adalah merupakan metode pengurutan data yang kompleks.

DAFTAR PUSTAKA

- Abdul kadir. 1987, *Pemrograman Pascal (Menggunakan Turbo Pascal 7.0 Borland Pascal 7.0)*, edisi pertama, Andi Offset, Yogyakarta.
- Neil W. Webre, Daniel F.Stubbs. 1985, *Data Structures with Abstract Data Types and Pascal*, California Polytechnic State University, San Luis Obisco.
- Santosa P. Insap. 1992., *Struktur Data menggunakan Turbo Pascal 6.0*, edisi pertama, Andi Offset Yogyakarta.
- Santosa P. Insap, 1990, *Dasar-Dasar Pemrograman Pascal, Teori dan Program Terapan*, edisi ketiga, Andi Offset, Yogyakarta.
- Tanenbaum, Aoron H, ; Augenstein, Hashe j, 1981, *Data Structure Using Pascal*, PrenticeHall International, Inc.