

# PENERAPAN ALGORITMA QUICK PIVOT SORT DAN RADIX SORT PADA DATA BUKU

Sabarudi Waruwu<sup>1</sup>, Abdul Sani Sembiring<sup>1</sup>, Matias Julyus. F. S<sup>1</sup>

Prodi Teknik Informatika STMIK Budi Darma, Medan, Indonesia  
Jl. Sisingamangaraja No. 338, Medan, Indonesia

## Abstrak

Pengurutan data atau sorting merupakan salah satu jenis operasi penting dalam pengolahan data. Masalah pengurutan merupakan masalah yang sering muncul dalam pemrograman komputer. Banyak pengurutan yang berbeda algoritma telah dikembangkan dan ditingkatkan untuk membuat pengurutan lebih cepat. Komparasi dilakukan dengan menghitung tingkat akurasi algoritma Quick Pivot Sort dan Radix Sort. Pada penelitian ini, kedua algoritma sorting yaitu algoritma quick pivot sort dan radix sort dibandingkan dengan algoritma pencocokan dan pencarian string. Kemudian, user melakukan pencarian string terhadap data yang telah diurutkan sebelumnya. Perbandingan yang dimaksud pada penelitian ini adalah bertujuan untuk menganalisis tingkat akurasi dari kedua algoritma sorting tersebut menggunakan algoritma pencarian string. Semakin cepat atau tidak berpengaruh sama sekali

Kata Kunci : Sorting, Pengurutan Data, Akurasi, Kesimpulan

## I. PENDAHULUAN

Dalam perkembangan teknologi, pengurutan data atau sorting merupakan salah satu jenis operasi penting dalam pengolahan data. Pengurutan data sangat penting digunakan, sehingga sampai saat ini telah banyak metode-metode pengurutan data dan mungkin akan tetap bermunculan metode –metode yang baru. Ada banyak metode pengurutan data antara lain : bubble sort, bidirectional bubble sort, selection sort, shakersort, insertion sort, inplace merge sort, double storage merge sort, comb sort 11, shell sort, heap sort, exchange sort, merge sort, quick sort, quick sort with bubblesort, enhance quicksort, fast quick sort, radix sort, swap sort, dan lain sebagainya

Quick Pivot Sort adalah algoritma pengurutan yang dikembangkan oleh Tony Hoare. Algoritma Quick Pivot Sort melakukan perbandingan sebanyak  $n \log n$  untuk mengurutkan data sebanyak  $n$ . Kondisi paling buruk dari jumlah perbandingan adalah  $n^2$ . Secara praktis, Quick Pivot Sort beroperasi lebih cepat dibanding algoritma dengan kompleksitas  $n \log n$  lainnya. Keuntungan dari penggunaan algoritma Quick Pivot Sort adalah cepat dan efisien, sedangkan kerugian yang dapat dialami adalah menghasilkan performa yang buruk apabila data sudah terurut.

Radix Sort merupakan salah satu algoritma Non-Comparasion Sort (pengurutan tanpa perbandingan). Proses yang dilakukan dalam metode ini adalah mengklasifikasikan/menyelesaikan data sesuai dengan kategori terurut yang tertentu, dan tiap kategori dilakukan pengklasifikasian lagi, dan seterusnya sesuai kebutuhan, kemudian sub kategori-kategori atau bagian-bagian dari kategori tersebut digabungkan kembali.

Pengurutan data memegang peranan penting yang banyak dipertimbangkan agar keseluruhan permasalahan (terutama mengenai pengolahan data) menjadi lebih baik dan lebih cepat untuk diselesaikan, sehingga menghasilkan data yang akurat. Masalah pengurutan merupakan masalah yang sering muncul dalam

pemrograman komputer. Banyak pengurutan yang berbeda algoritma telah dikembangkan dan ditingkatkan untuk membuat pengurutan lebih cepat. Algoritma pengurutan adalah algoritma yang menyimpan suatu list pada suatu urutan tertentu, biasanya membesar atau mengecil, dan digunakan untuk mengurutkan angka ataupun huruf, proses pengurutan data pada penelitian skripsi ini membandingkan algoritma Quick Pivot Sort dan Radix Sort untuk mengurutkan angka.

## II. TEORITIS

### A. Quick Pivot Sort

Quicksort merupakan Algoritma pengurutan yang dikembangkan oleh Tony Hoare. performa rata-rata pengurutan  $O(n \log n)$  untuk mengurutkan  $n$  item[6]. Algoritma ini juga dikenal sebagai Partition-Exchange Sort atau disebut sebagai Sorting Pergantian Pembagi[6].

Quicksort merupakan Algoritma Pembagi. Pertama quicksort membagi list yang besar menjadi dua buah sub list yang lebih kecil: element kecil dan element besar. Quicksort kemudian dapat menyortir sub list itu secara rekursif. Langkah-Langkah pengerjaannya ialah[6]:

1. Ambil sebuah elemen, yang disebut dengan pivot, pada sebuah daftar.
2. Lakukan pemindaian list pada elemen dengan nilai yang kecil dari pivot berada sebelum pivot, sedangkan seluruh element yang memiliki nilai yang lebih besar dari pivot berada setelahnya (nilai yang sama dapat berada pada pivot setelahnya). Setelah pemisahan, pivot berada pada posisi akhirnya. Operasi ini disebut Partition.
3. Sub list kemudian disortir secara rekursif dari elemen yang lebih kecil dan sub list dari elemen yang lebih besar.

### B. Radix Sort

Radix Sort adalah metode sorting tanpa perbandingan dengan kata lain, sorting Non-Comparasion sort dimana dalam prosesnya tidak melakukan perbandingan antar data. Secara umum proses

yang dilakukan dalam metode ini adalah mengklasifikasikan data sesuai dengan kategori tertentu yang tertentu dan dalam tiap kategorinya dilakukan pengklasifikasian lagi dan seterusnya sesuai dengan kebutuhan. Dan kemudian subkategori-subkategori tersebut digabungkan kembali, yang dilakukan hanya dengan metode sederhana concatenation[2].

Proses dasar Radix Sort adalah mengkategorikan data-data menjadi sub kumpulan data sesuai dengan nilai radix-nya, mengkonkatenasinya, kemudian mengkategorikannya kembali berdasar nilai radix lainnya[2].

### C. Data

Secara umum, pengertian data dapat didefinisikan sebagai nilai (value) yang mempresentasikan deskripsi dari suatu objek atau peristiwa. Data di bentuk dari data mentah (raw data) yang berupa angka, karakter, gambar, atau bentuk lainnya. Data dapat diartikan sebagai sesuatu yang diketahui atau yang dianggap atau anggapan.

## III. ANALISA

### A. Analisa Masalah

Dalam pengolahan data, pengurutan (sorting) sudah menjadi sesuatu yang penting untuk mempermudah dalam menyusun data. Pengurutan adalah proses menyusun kembali data yang sebelumnya masih acak kedalam urutan tertentu. Permasalahan pengurutan data dipengaruhi oleh beberapa faktor yaitu banyaknya data yang diurutkan, kapasitas memori dan kemampuan dari media penyimpanan. Pemilihan algoritma sangat ditentukan oleh struktur data yang digunakan. Data yang digunakan juga bervariasi, baik dalam jumlah data maupun jenis data yang akan diurutkan. Tidak ada algoritma yang terbaik dalam masalah pengurutan (sorting). Algoritma quick pivot sort merupakan algoritma pengurutan dengan kinerja yang cepat. Algoritma quick pivot sort membandingkan sebanyak  $n \log n$  untuk mengurutkan data sebanyak  $n$ . Sementara algoritma radix sort adalah algoritma non-comparison sort dimana dalam proses pengurutan tidak ada perbandingan setiap string seperti algoritma quick pivot sort.

Pada penelitian ini, penulis ingin menguji algoritma sorting dengan mengurutkan judul buku dan menguji algoritma pengurutan mana yang memiliki tingkat akurasi yang lebih cepat. Pengurutan dari algoritma quick pivot sort dan radix sort dapat dilihat dari tahapan-tahapan berikut ini.

### B. Penerapan Proses Sorting Quick Pivot Sort

Tahapan dalam melakukan partisi pada algoritma *quick pivot sort* adalah sebagai berikut:

1. Pilih  $x \in \{A_1, A_2, \dots, A_n\}$  sebagai pivot
2. Lakukan pemindaian *list* dari kiri ke kanan sampai ditemukan  $A_p > x$
3. Lakukan pemindaian *list* dari kanan ke kiri sampai ditemukan  $A_q < x$
4.  $Swap A_p \rightarrow \leftarrow A_q$
5. Ulangi langkah ke 2, sampai kedua pemindaian bertemu ditengah.

Label Buku= 046, 049, 053, 050, 075, 065

046	049	053	049	075	065
-----	-----	-----	-----	-----	-----

Penyelesaian:

#### 1. Menentukan pivot (x)

A1	A2	A3	A4	A5	A6
046	049	053	050	075	065
			x		

2. Lakukan pemindaian *list* dari kiri ke kanan sampai ditemukan  $A_p \geq x$  dan pemindaian *list* dari kanan ke kiri sampai ditemukan  $A_q \leq x$ .

A1	A2	A3	A4	A5	A6
046	049	053	050	075	065
p			x		q

#### 3. $Swap A_p \rightarrow \leftarrow A_q$

A1	A2	A3	A4	A5	A6
046	049	053	050	075	065
	p		x	q	

4. Ulangi langkah ke 2 dari posisi  $p+1$  dan dari posisi  $q-1$ , sampai kedua pemindaian bertemu ditengah *list*.

A1	A2	A3	A4	A5	A6
046	049	053	050	075	065
		p	x		q

#### 5. $Swap A_p \rightarrow \leftarrow A_q$

A1	A2	A3	A4	A5	A6
046	049	053	050	075	065
			x		
			q		
			p		

#### 6. $Swap A_p \rightarrow \leftarrow A_q$

A1	A2	A3	A4	A5	A6
046	049	050	053	075	065
	p		x		q

7. Ulangi langkah 2 dari posisi  $p+1$  dan dari posisi  $q-1$ , sampai kedua pemindaian bertemu ditengah *list*.

A1	A2	A3	A4	A5	A6
046	049	050	053	075	065
		p	x	q	

8. Proses berhenti karena  $p \leq q$  berada ditengah *list*.

A1	A2	A3	A4	A5	A6
046	049	050	053	075	065
			x		
			q		
			p		

#### 9. Menentukan kembali pivot (x)

A1	A2	A3	A4	A5	A6
046	049	050	053	075	065
			p	x	q

10. Proses berhenti karena  $p \leq q$  berada ditengah *list*.

A1	A2	A3	A4	A5	A6
046	049	050	053	075	065
				x	
				q	
				p	

11. Hasil akhir keseluruhan proses.

A1	A2	A3	A4	A5	A6
046	049	050	053	065	075

### C. Penerapan Sorting Radix Sort

Contohnya adalah pengurutan sebuah kumpulan angka (indeks buku= 046, 049, 053, 049, 075, 065). Tahapan dari pengurutan algoritma radix sort adalah sebagai berikut:

1. Data dibagi sesuai digit paling kanan

046	049	053	050	075	065
-----	-----	-----	-----	-----	-----

Kategori digit	Isi
0	050
1	-
2	-
3	053
4	-
5	075,065
6	046
7	-
8	-
9	049

2. Hasil pengkategorian tersebut lalu digabungkan kembali dengan metode konkatenasi, menjadi:

050	053	075	065	046	049
-----	-----	-----	-----	-----	-----

3. Kemudian pengkategorian dilakukan kembali, namun kali ini berdasar digit kedua atau digit tengah, dan perhatikan bahwa urutan pada tiap sub kumpulan data harus sesuai dengan urutan kemunculan pada kumpulan data.

050	053	075	065	046	049
-----	-----	-----	-----	-----	-----

Kategori digit	Isi
0	-
1	-
2	-
3	-
4	046,049
5	050,053
6	065
7	075
8	-
9	-

4. Dan kemudian kongkatenasikan kembali, yang merupakan hasil akhir dari pengurutan berdasarkan radix sort.

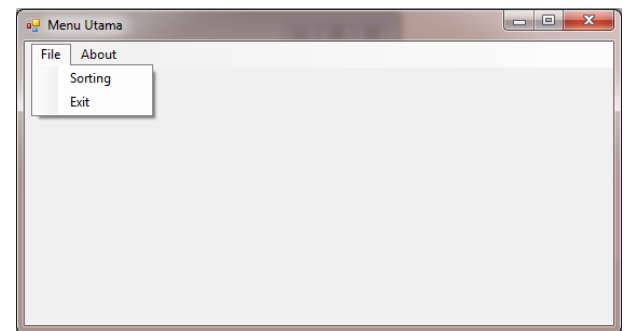
046	049	050	053	065	075
-----	-----	-----	-----	-----	-----

Jadi, proses pengurutan data dengan menggunakan algoritma radix sort untuk data integer tersebut, berhenti pada tahap ke 4. Sebagaimana telah dijelaskan pada bab sebelumnya, algoritma radix sort mengelompokkan data berdasarkan digit yang memiliki significant position dan value yang sama

Dari hasil pengurutan quick pivot sort dan radix sort pada kode barang, kita dapat melihat bahwa radix sort memiliki tingkat akurasi yang lebih tinggi daripada quick pivot sort dalam melakukan pengurutan. Dimana radix sort hanya melakukan 4 tahapan, sementara quick pivot sort melakukan 11 tahapan. Maka daripada itu dapat ditentukan hasil pengurutan menggunakan algoritma radix sort lebih cepat dari algoritma quick pivot sort

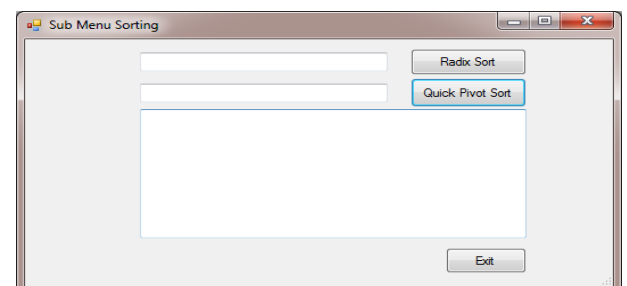
### IV. IMPLEMENTASI

Tampilan program merupakan interface antara user dengan aplikasi komparasi yang telah dibuat. Tampilan awal aplikasi yang telah dibuat terdiri dari tampilan menu utama yang berfungsi untuk memberi user kesempatan memilih searching atau exit, tampilan searching untuk melakukan psroses, tampilan proses untuk menampilkan Tampilan Menu Utama. Tampilan menu utama dapat dikatakan sebagai antar muka (user interface) antara user dengan aplikasi. Menu utama menampilkan pilihan menu yang tersedia pada aplikasi. Pada menu utama tersedia 2 (dua) pilihan menu yaitu File dan About. Adapun screenshoot gambar untuk tampilan menu utama dapat dilihat pada gambar 1.



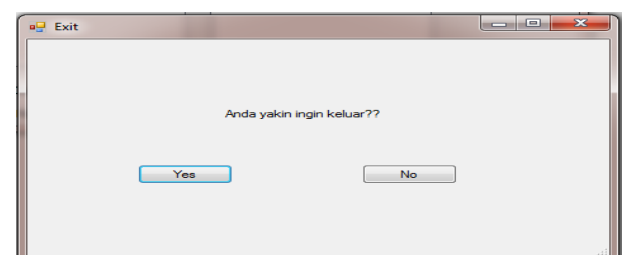
Gambar 1. Menu Utama

Tampilan Sub menu sorting merupakan tampilan yang disediakan untuk menampilkan tombol quick pivot sort dan radix sort. Adapun screenshoot gambar untuk tampilan Tampilan Sub menu sorting dapat dilihat pada gambar 2.



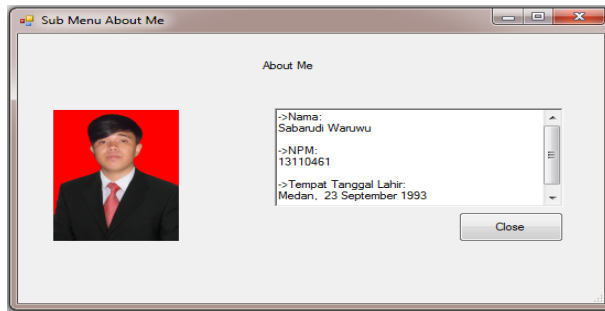
Gambar 2. Tampilan Sub menu sorting

Tampilan sub menu exit digunakan untuk keluar oleh user. Adapun screenshoot gambar untuk tampilan tampilan arti istilah psikologi dapat dilihat pada gambar 3.



Gambar 3. Tampilan Menu Exit

Tampilan menu about akan menampilkan tentang biodata si pembuat aplikasi. Tampilan sub menu about dapat dilihat pada gambar 4. di bawah ini.



Gambar 4. Tampilan sub menu About me

## V. KESIMPULAN

Berdasarkan hasil akhir pemecahan masalah maka penulis dapat menarik sebuah kesimpulan, dimana kesimpulan tersebut nanti dapat kiranya berguna bagi pembaca, sehingga penulisan skripsi ini dapat lebih berguna dan bermanfaat. Adapun kesimpulan-kesimpulan tersebut antara lain sebagai berikut :

1. Implementasi algoritma quick pivot sort yang digunakan dapat mempercepat proses pengurutan.
2. Implementasi algoritma radix sort yang digunakan aplikasi untuk dibandingkan dengan algoritma quick pivot sort.
3. Penerapan Algoritma Quick Pivot Sort dan Radix Sort dapat digunakan dalam mengurutkan data buku.

## REFERENCES

- [1] Audy, "Komparasi Algoritma Quicksort dan Bucket Sort pada Pengurutan Data Integer," *ULTIMATICS*, vol. VII, no. 1, pp. 7-13, 2015.
- [2] S. Lahera dan N. Shah, "RADIX SORT WITH REDUCED TIME COMPLEXITY," *International Journal of Students Research in Technology & Management*, vol. 1, no. 6, pp. 624-626, 2013.
- [3] A. Sonita dan F. Nurtaneo, "ANALISIS PERBANDINGAN ALGORITMA BUBBLE SORT, MERGE SORT, DAN QUICK SORT DALAM PROSES PENGURUTAN KOMBINASI ANGKA DAN HURUF," *Jurnal Pseudocode*, vol. 2, no. 2, pp. 75-80, 2015.
- [4] U. Patel dan M. Thakkar, "Survey on Exact Pattern Matching Algorithm," *International Journal for Scientific Research & Development*, vol. 1, no. 8, pp. 1623-1626, 2013.
- [5] Ritayani, "PENGANTAR ALGORITMA DAN PEMROGRAMAN," Fakultas Ilmu Komputer Universitas Almuslim, Aceh, 2012.
- [6] R. Mansi, "Enhanced Quicksort Algorithm," *The International Arab Journal of Information Technology*, vol. 7, no. 2, pp. 161-166, 2010.
- [7] K. Al-Khamaiseh dan S. ALShagrin, "A Survey of String Matching Algorithms," *International Journal of Engineering Research and Applications*, vol. 4, no. 7, pp. 144-156, 2014.
- [8] J. Bhandari dan A. Kumar, "STRING MATCHING RULES USED BY VARIANTS OF BOYER-MOORE ALGORITHM," *Journal of Global Research in Computer Science*, vol. 5, no. 1, pp. 8-11, 2014.
- [9] Haviluddin, "Memahami Penggunaan UML (Unified Modelling Language)," *Informatika Mulawarman*, vol. 6, no. 1, pp. 1-15, 2011.
- [10] Neti, "PERANCANGAN APLIKASI RENTAL MOBIL PADA CV KARYA BERSAMA PALEMBANG," *MDP*, pp. 1-10, 2008.