ANALISIS KEBUTUHAN WAKTU ALGORITMA *INSERTION SORT*, *MERGE SORT*, DAN *QUICK SORT* DENGAN KOMPLEKSITAS WAKTU

SKRIPSI

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
Sarjana Sains Bidang Studi Matematika

Oleh
M. ALLBAR PRATAMA
NIM 08121001053

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SRIWIJAYA
JULI 2017

LEMBAR PENGESAHAN

ANALISIS KEBUTUHAN WAKTU ALGORITMA *INSERTION SORT*, *MERGE SORT*, DAN *QUICK SORT* DENGAN KOMPLEKSITAS WAKTU

SKRIPSI

Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Sains Bidang Studi Matematika

Oleh:

M. ALLBAR PRATAMA
NIM 08101001053

Indralaya, Juli 2017

Pembimbing Pembantu	Pembimbing Utama
Hj. Irmeilyana, M.Si.	Anita Desiani, M.Kom.
NIP 19740517 1999 03 2 003	NIP 19771211 2003 12 2 002

Mengetahui,
Ketua Jurusan Matematika

Drs. Sugandi Yahdin, M.M.
NIP 195807271986031003

LEMBAR PERSEMBAHAN

Motto :

"Iqro'(Bacalah)"
(Al- 'Alaq: 1)

"Ilmu ada tiga tahapan. Jika seseorang memasuki tahap pertama, dia akan sombong. Jika dia memasuki tahap kedua, ia akan tawadu' (rendah hati). Dan jika memasuki tahapan ketiga, dia akan merasa dirinya tidak ada apa-apanya."
(Umar bin Khattab)

"Sticks and stones may break my bones. But words, will never hurt me."

Skripsi ini Kupersembahkan kepada :

- *Bapak dan Ibu tercinta*
- *Adik-Adikku yang aku sayangi*
- *Dosen dan Guruku*
- *Sahabat-Sahabat Terbaikku, dan*
- *Almamater Kebanggaanku*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur kehadirat Allah SWT Yang Maha Pengasih lagi Maha Penyayang atas segala limpahan rahmat, karunia, serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini sesuai dengan waktu yang telah direncanakan dengan judul **"ANALISIS KEBUTUHAN WAKTU ALGORITMA *INSERTION SORT*, *MERGE SORT*, DAN *QUICK SORT* DENGAN KOMPLEKSITAS WAKTU"**. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Besar kita Nabi Muhammad SAW beserta seluruh keluarga dan sahabatnya yang telah membawa kita dari jaman kebodohan ke jaman yang terang benderang.

Dengan penuh rasa hormat, cinta, kasih sayang dan kerendahan hati, penulis mempersembahkan skripsi ini khusus untuk kedua orang tua tercinta, terkasih dan tersayang **Ayah Agus Herman** dan **Ibu Sri Anita** yang telah merawat dan mendidik penulis dengan penuh rasa cinta dan kasih sayang, serta dukungan yang sangat berharga berupa motivasi keluarga, do'a, perhatian, semangat, serta material untuk penulis selama ini.

Penulis menyadari bahwa keberhasilan penyusunan skripsi ini tidak terlepas dari bantuan pembimbing, dan berbagai pihak lain baik langsung maupun tidak langsung. Dalam kesempatan ini, penulis ingin menyampaikan rasa terimakasih yang besar kepada :

1. Ibu **Anita Desiani, M.Kom** selaku Dosen Pembimbing Utama yang telah bersedia meluangkan waktu dengan penuh kesabaran dan perhatian dalam

memberikan banyak ide pemikiran, bimbingan, nasehat, pengarahan, serta kritik dan saran yang sangat berguna bagi penulis selama pengerjaan skripsi sehingga skripsi ini dapat terselesaikan dengan baik dan sesuai dengan waktu yang direncanakan.

2. Ibu **Hj. Irmeilyana, M.Si** selaku Dosen Pembimbing Pembantu sekaligus Dosen Pembimbing Akademik penulis yang juga telah banyak memberikan arahan dan bimbingan kepada penulis selama pengerjaan skripsi ini maupun selama belajar di Jurusan Matematika FMIPA Universitas Sriwijaya.

Selain itu, penulis juga mendapatkan dukungan dari pihak-pihak lain selama masa perkuliahan hingga penyelesaian skripsi ini. Oleh karena itu, penulis juga ingin mengucapkan terima kasih kepada :

1. Bapak **Drs. Sugandi Yahdin, M.M.** selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sriwijaya dan Dosen Pembahas yang telah bersedia menyediakan waktu yang selalu memberikan arahan dan masukan dalam proses pengerjaan skripsi ini.
2. Bapak **Drs. Putra Bahtera Jaya Bangun, M.Si** yang sebelumnya menjabat sebagai Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sriwijaya dan Dosen Pembahas yang telah bersedia menyediakan waktu yang selalu memberikan arahan dan masukan dalam proses pengerjaan skripsi ini.
3. Bapak **Drs. Robinson Sitepu, M.Si** dan Ibu **Indrawati, M.Si** selaku Dosen Pembahas Skripsi yang telah bersedia meluangkan waktunya dalam

memberikan tanggapan, kritik serta saran yang bermanfaat dalam perbaikan penyelesaian skripsi ini.

4. Seluruh **Dosen di Jurusan Matematika FMIPA Universitas Sriwijaya, Guru-Guru penulis di SMA Negeri 13 Palembang, SMPN 52 Palembag, dan SDN 136 Palembang.** Terima kasih atas ilmu yang telah Bapak/Ibu berikan untuk penulis selama proses pendidikan.
5. Seluruh **Staf dan Dosen** di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sriwijaya terima kasih atas bimbingan dan bantuan kepada penulis.
6. Adik-Adik yang aku sayangi **M. Allkap Dwi Aditya** dan **Regina Herman Arrayan Sita,** terima kasih atas semangat, do'a, motivasi, dan dukungan yang telah diberikan.
7. Sahabat-sahabat angkatan 2012: **Reyfaldo Tommy, Ismail, Ahmad Junaidi, Rakhatama Gusri, Atoihillah Abdullatif, Akbar Yulanda, Muhamad Ario, Ferdy Octariansyah, M. Edwin Afriansyah, Rudi Gunawan, M. Ari Juliansyah, M. Syahisal Arialdi, Bastruman, Chandra Gunawan, Dian Permatasari, Tita Adeasty, Mirza Denia Putri, Triyani, Dewi Rakhmatia, Icha Puspita, Elvia, Risfa Risa, Emi Widart, Nelda Amelia, Defita Yulanda, Amira Fitri Adila, Lucky Sri Dorce** dan sahabat-sahabat seangkatan yang lainnya terima kasih untuk semua canda tawa, suka duka, semangat, dan nasehat.
8. Kakak Tingkat 2010 **Hendriansyah, Habib Amaluddin, Januar Dwi Admaja, Toni Kesumajati, Muhammad Darmawan, Mgs. Fairuz, Udo**

Febrian, Repi Minorilistita, Fani Maulia Rahmah, Dian, adik tingkat **Aldyo Eka Putra, Salman Al-Farisy dan adik-adik tingkat yang lainnya** terima kasih atas segala bentuk bantuan dan canda tawa yang telah diberikan.

9. Teman-teman SMP **Adi Pribadi, Adi Mandala, Faip Mardoni, M. Rasyid Ridho, M. Rico Trijunarsyah, Akhmad Khairl Alam, Didit Pringgondai** dan Ketua Voli Vigar **Rudi Rahman** terima kasih atas perhatian, bantuan, nasehat, segala suka duka dan dukungannya**.**
10. Semua pihak yang tidak dapat penulis sebutkan satu-persatu yang telah memberikan do'a, dukungan dan masukan yang berguna untuk menyelesaikan skripsi ini. Semoga segala kebaikan dan pertolongan semuanya mendapatkan berkah dari Allah SWT.

Akhir kata dengan segala kerendahan hati, penulis mohon maaf apabila masih banyak kekurangan dalam penyusunan skripsi ini dan masih jauh dari kesempurnaan, maka saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Semoga skripsi ini dapat menambah pengetahuan dan bermanfaat bagi semua pihak yang memerlukan. Amin.

Wassalamu'alaikum Wr. Wb

Indralaya, April 2017

Penulis

ANALYZING TIME REQUIREMENTS OF *INSERTION SORT, MERGE SORT,* AND *QUICK SORT* ALGORITHMS WITH TIME COMPLEXITY

By

M. Allbar Pratama

08121001053

ABSTRACT

Sorting is a crucial problem in data processing or database. Data processing will be more simple if the data has been sorted. Sorting problem requires special techniques to make the process of sorting faster. The techniques are named as sorting algorithms. The reliability of an algorithm can be measured by its time complexities. The time complexity $T(n)$ is the number of operations performed in an algorithm for N data input. One of time complexities is *Big-O* or worst case. The Worst case (Big-O) is a time complexities for the worst condition of an algorithm. This study will analyze the time complexity of the algorithms Insertion Sort, Merge Sort and Insertion Sort based on their Big-O (worst case). Each algorithm will be calculated its complexity time in two ways. The first is calculated based on their steps in sorting process and the second is calculated based on their coding and running program using *C++*. The time complexity of Merge Sort is $O(n \log n)$ and time complexity of Quick Sort and Insertion Sort is $O(n^2)$, it means the time complexity of Merge Sort is less and faster for large *N* data input than Quick Sort and Insertion Sort. Otherwise Insertion Sort is faster for small *N* data input than Merge Sort and Quick Sort. Quick sort needs much time to sort data not only for small *N* data input but also for large *N* data input. It means Quick Sort doesn't work well in worst case.

Keyword: *Sorting, Insertion Sort, Quick Sort, Merge Sort, Time Complexity, Worst Case, Big-O*

ANALISIS KEBUTUHAN WAKTU ALGORITMA *INSERTION SORT, MERGE SORT,* DAN *QUICK SORT* BERDASARKAN KOMPLEKSITAS WAKTU

Oleh
M. Allbar Pratama
08121001053

ABSTRAK

Pengurutan merupakan salah satu hal yang penting dalam pengolahan data atau basis data karena dalam pemrosesan data akan lebih mudah jika data yang diolah telah terurut. Dalam masalah pengurutan dibutuhkan teknik pengurutan yang dapat membantu proses pengurutan lebih cepat yang disebut dengan algoritma *sorting*. Kehandalan suatu algoritma dapat diukur dengan mengukur kompleksitas waktu suatu algoritma. Kompleksitas waktu $T(n)$ adalah jumlah operasi yang dilakukan dalam suatu algoritma dari masukan sebanyak n. Salah satu pengukuran kompleksitas waktu algoritma adalah *Big-O* atau *worst case* (kondisi terburuk). *Worst case* (*Big-O*) merupakan pengukuran waktu untuk kondisi terburuk dari suatu algoritma. Dalam penelitian ini, dianalisis komplesitas waktu dari masing algoritma *Insertion Sort, Merge Sort*, dan *Quick Sort* berdasarkan *Big-O* atau kondisi terburuknya. Masing-masing algoritma dihitung waktu kompleksitasnya dengan dua cara. Pertama, berdasarkan langkah-langkah masing-masing algoritma dan cara kedua dihitung berdasarkan *coding* dan implementasi program dari tiap-tiap algoritma menggunakan bahasa pemograman *C++*. Kompleksitas waktu dari *Merge Sort* adalah $O(n \log n)$ dan kompleksitas waktu dari *Quick Sort* dan *Insertion Sort* adalah $O(n^2)$, artinya kompleksitas waktu dari *Merge Sort* lebih cepat untuk ukuran *input* data n besar daripada *Quick Sort* dan *Insertion Sort*. Sedangkan *Insertion Sort* lebih cepat untuk *input* data n kecil daripada *Merge Sort* dan *Quick Sort*. *Quick Sort* membutuhkan banyak waktu untuk mengurutkan data baik untuk *input* data n kecil maupun n besar. Artinya *Quick Sort* tidak bekerja dengan baik untuk kasus *worst case*.

Kata Kunci: *Sorting, Insertion Sort, Quick Sort, Merge Sort, Kompleksitas Waktu, Worst Case, Big-O*

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pengurutan (*sorting*) merupakan salah satu algoritma yang sangat penting dalam dunia Teknologi Informasi terutama pada bagian pengolahan data ataupun basis data (Munir, 2011). Dalam pengolahan data, misalnya pembacaan, akan lebih mudah dilakukan apabila data yang diolah sudah terurut. Oleh karena itu dibutuhkan algoritma pengurutan untuk mengurutkan data. Adanya kebutuhan terhadap proses pengurutan memunculkan bermacam-macam algoritma pengurutan. Menurut Canaan, dkk. (2011), algoritma pengurutan yang populer antara lain: *Bubble Sort, Selection Sort, Insertion Sort, Heap Sort, Shell Sort, Qucik Sort, Merge Sort, Radix Sort,* dan *Tree Sort.*

Data yang sudah terurut memiliki beberapa keuntungan. Selain mempercepat waktu pencarian, dari data yang terurut dapat langsung diperoleh nilai maksimum dan nilai minimum. Misalnya untuk data numerik yang terurut menurun, nilai maksimum adalah elemen pertama *array*, dan nilai minimum adalah elemen terakhir *array* (Munir, 2011).

Banyaknya algoritma pengurutan menunjukkan bahwa persoalan pengurutan adalah persoalan yang kaya dengan solusi algoritmik. Sehingga seorang *programmer* harus menentukan algoritma pengurutan manakah yang membutuhkan kebutuhan waktu algoritma paling sedikit. Kebutuhan waktu suatu algoritma berguna dalam menentukan kinerja suat algoritma, sehingga diperlukan kriteria formal yang

digunakan untuk menilai algoritma yang terbaik. Kriteria tersebut disebut dengan kompleksitas waktu.

Kompleksitas waktu terdiri dari *best case, worst case,* dan *average case* merupakan hal penting untuk mengukur efisiensi suatu algoritma. Kompleksitas waktu dari suatu algoritma yang terukur dianggap sebagai suatu fungsi ukuran masukan (Nugraha, 2012). Kompleksitas waktu diekspresikan sebagai jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan *n*. Dengan menggunakan kompleksitas waktu, laju peningkatan waktu yang diperlukan algoritma dapat ditentukan sesuai dengan meningkatnya ukuran masukan *n* (Munir, 2005).

Menurut Nugraha (2012) analisis kompleksitas waktu algoritma adalah membandingkan waktu yang dibutuhkan algoritma dalam menyelesaikan perintah. Estrada (2003) juga menyebutkan dengan menganalisis kompleksitas waktu disimpulkan bahwa banyaknya jumlah data-*n* berpengaruh terhadap kebutuhan waktu yang diperlukan. Pada penelitian Sonita dan Nurtaneo (2015) dari hasil analisis perbandingan Algoritma *Bubble Sort, Quick Sort*, dan *Merge Sort* disimpulkan bahwa Algoritma *Quick Sort* memiliki kompleksitas waktu yang lebih cepat daripada *Bubble Sort* membutuhkan waktu yang paling lama untuk rata-rata waktu yang dibutuhkan (*average case*).

Average case dalam kompleksitas waktu pada pengurutan kasusnya digunakan untuk men-*sorting* masalah pengurutan dengan kondisi yang posisi urutan datanya acak atau inputan datanya tidak teratur. Untuk kondisi masalah *sorting* yang terbalik *sorting* yang digunakan adalah *worst case* (waktu terburuk) atau disebut juga

sebagai *O*-Besar (*Big-O*). Pada penelitian ini, algoritma *Insertion Sort*, *Merge Sort*, dan *Quick Sort Sort* diteliti kompleksitas waktunya untuk mengetahui algoritma mana yang paling efisien dari ketiga algoritma tersebut berdasarkan *worst case*-nya.

1.2. Perumusan Masalah

Masalah yang dibahas dalam penelitian ini adalah:

1. Bagaiamana menentukan kompleksitas waktu dari algoritma *Insertion Sort*, *Merge Sort*, dan *Quick Sort* dengan menyelesaikan masalah pengurutan berdasarkan *worst case*.
2. Bagaimana menentukan algoritma yang paling cepat dalam menyelesaikan masalah pengurutan berdasarkan *worst case*.

1.3. Pembatasan Masalah

Batasan dalam penelitian ini adalah:

1. Algoritma *sorting* yang diteliti hanya yang menggunakan *array* dan tidak meneliti algoritma yang menggunakan *linked list.*
2. Perhitungan kompleksitas waktu algoritma tidak melihat spesifikasi detail dari komputer yang digunakan.
3. Komplekstias waktu yang dihitung hanya kompleksitas waktu yang terburuk saja (*worst case*).

1.4. Tujuan

Tujuan penelitian ini adalah:

1. Menentukan kompleksitas waktu dari algoritma *Insertion Sort*, *Merge Sort*, dan *Quick Sort* dengan menyelesaikan masalah pengurutan berdasarkan *worst case*.
2. Menentukan algoritma yang paling cepat dalam menyelesaikan masalah pengurutan berdasarkan *worst case*.

1.5. Manfaat

Manfaat dalam penelitan ini adalah:

1. Dapat memperoleh pemahaman terhadap penggunaan Algoritma *Insertion Sort, Merge sort,* dan *Quick Sort* dalam men-*sorting* data.
2. Dapat menggunakan algoritma yang paling efisien dalam men-*sorting* data sehingga dapat mempersingkat waktu untuk eksekusi program.

DAFTAR PUSTAKA

Aho. Alfred, V. Sethi. Ravi. Ullman. Jeffrey, D. 1986. *Compilers: Principles, Techniques, and Tools.* Addison-Wesley.

Astuti, Y.D. 2005. *Logika dan Algoritma.* Bandung: Universitas Gunadarma.

Azizah, U.N. 2013. Perbandingan Detektor Tepi Prewit dan Detektor Tepi Laplacian Berdasarkan Kompleksitas Waktu dan Citra Hasil. *Jurnal UPI*.

Azmoodeh, Manoocher. 1988. *Abstract Data Types and Algorithms*. Macmillan Education.

Canaan, C. Garai, M. S. dan Daya, M. 2011. *Popular Sorting Algorithms. World Applied Programming.* Volume 1, No. 1. 62-71

Estrada S., A.H. 2003. Telaah Waktu Eksekusi Program Terhadap Kompleksitas Waktu Algoritma *Brute Force* dan *Divide and Conquer* dalam Penyelesaian Operasi *List*. *Jurnal Ilmu Komputer dan Teknologi Informasi Institut Teknologi Bandung.* Volume 3, No. 2.

Fatta, Hanif A. 2006. *Dasar Pemrograman C++ Disertai dengan Pengenalan Pemrograman Berorientasi Objek*. Andi Publisher.

Levitin, Anany. 2007. *Introduction to The Design and Analysis of Algorithms.* Addison-Wesley.

Mehta, D. P. dan Sahni, S. 2005. *Handbook of Data Structures and Applications*. Chapman & Hall/CRC Computer and Information Science Series, United States of America.

Munir, Rinaldi. 2005. *Matematika Diskrit*. Bandung: Informatika Bandung.

Munir, Rinaldi. 2011. *Algoritma & Pemrograman Dalam Bahasa Pascal dan C.* Bandung: Informatika Bandung.

Nugraha, D. W. 2012. Penerapan Kompleksitas Waktu Algoritma Prim untuk Menghitung Kemampuan Komputer Dalam Melaksanakan Perintah. *Jurnal Ilmiah Foristek Universitas Taduloka.* Volume 2, No. 2. 195-202

Palui, S., dan Gupta, S. 2014. A Unique Sorting Algorithm with Linear Time & Space Complexity. *International Journal of Research in Engineering and Technology*. Volume 3, No. 11. 264-268

Raharjo, Budi. 2005. *Teknik Pemprograman Pascal*. Bandung: Informatika Bandung.

Sedgewick, Robert. 1992. *Algorithms in C++*. Addison-Wesley.

Sonita, A. dan Nurtaneo, F. 2015. Analisis Perbandingan Algoritma *Bubble Sort*, *Merge Sort*, dan *Quick Sort* dalam Proses Pengurutan Kombinasi dan Huruf. *Jurnal Pseudocode Universitas Muhammadiyah Bengkulu*. Volume 2, No. 2. 75-80