

APLIKASI PENGHITUNG KOMPLEKSITAS ALGORITMA

Rosita Kartikasari¹, Fazmah Arief Yulianto², Dhinta Darmantoro³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Aplikasi komputer banyak digunakan sebagai alat bantu dalam berbagai hal. Aplikasi yang digunakan tersebut diharapkan dapat bekerja secara cepat dalam melakukan prosesnya. Aplikasi yang bekerja cepat, harus dibangun oleh algoritma yang efisien, yang diukur dari segi waktu (running time) dan ruang (memory). Semakin kecil waktu dan ruang yang dibutuhkan, maka semakin baik algoritma tersebut. Besaran pengukuran waktu atau ruang algoritma disebut kompleksitas algoritma.

Untuk dapat mengetahui dan membandingkan efisiensi suatu algoritma dengan algoritma yang lainnya, perlu dilakukan perhitungan kompleksitasnya terlebih dahulu. Perhitungan kompleksitas algoritma secara manual, membutuhkan waktu yang cukup banyak. Terutama jika algoritma yang dihitung dan dibandingkan cukup panjang dan banyak jumlahnya. Oleh karena itu, dibutuhkan suatu aplikasi yang dapat membantu dalam penghitungan kompleksitas algoritma untuk memilih algoritma yang lebih efisien untuk membangun suatu aplikasi komputer.

Dalam tugas akhir ini, pembuatan aplikasi dilakukan dengan menggunakan teknik kompilasi dan bahasa pemrograman Borland Delphi 7.

Hasil pengujian menunjukkan bahwa, penghitungan kompleksitas algoritma dapat dilakukan dengan Aplikasi Penghitung Kompleksitas Algoritma, dengan hasil mendekati teori kompleksitas asimptotik.

Kata Kunci : Algoritma, Kompleksitas Algoritma, Borland Dephi 7.

Abstract

Computer application a lot of used as an assistive appliance in so many matter. Application used the expected can work quickly in doing its process. Laboring application quickly, have to be developed by efficient algorithm, what is measured from time facet (running time) and space (memory). Smaller require space and time, hence progressively the algorithm goodness. Asses of measurement of time or algorithm space referred as an algorithm complexity.

To can know and compare the efficiency of an algorithm with the other one, require to be done by its complexity calculation beforehand. In manual algorithm complexity calculation, need a lot of time. Especially if which it algorithm calculated and compared to long enough and a lot of its amount. Therefore, require by an application which can assist in enumeration of algorithm complexity to chosen the more efficient algorithm to develop a computer application.

In this final project, build application by using language of programming Borland Delphi 7. The result of testing showed, algorithm complexity calculation can use Application of Complexity Algorithm, which the results empirics with theory of asymptotic complexity.

Keywords : Algorithm, Algorithm Complexity, Borland Delphi 7

BAB I

PENDAHULUAN

1.1 Latar belakang

Komputer merupakan suatu perangkat elektronika yang dapat menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis dengan aturan tertentu. Saat ini, telah banyak hal yang menggunakan alat bantu aplikasi komputer, baik untuk membantu pekerjaan, untuk hiburan, kebutuhan komersil, bahkan untuk ilmu pengetahuan.

Aplikasi komputer yang digunakan tentu saja diharapkan dapat bekerja cepat dalam melakukan prosesnya. Bagi seorang *programmer*, untuk membuat aplikasi yang bekerja cepat, harus memikirkan dan membuat algoritma yang efisien, yang diukur dari segi waktu (*running time*) dan ruang (memori). Semakin kecil waktu dan ruang yang dibutuhkan, maka semakin baik algoritma tersebut. Besaran pengukuran waktu atau ruang algoritma disebut kompleksitas algoritma.

Untuk dapat mengetahui dan membandingkan efisiensi suatu algoritma dengan algoritma yang lainnya, perlu dilakukan perhitungan kompleksitasnya terlebih dahulu. Perhitungan memori yang dibutuhkan dilihat dari struktur data yang digunakan, sedangkan perhitungan waktu yang dibutuhkan dilihat dari banyaknya perintah atau operasi yang dieksekusi.

Perhitungan kompleksitas algoritma secara manual, membutuhkan waktu yang cukup banyak. Terutama jika algoritma yang dihitung dan dibandingkan cukup panjang dan banyak jumlahnya. Oleh karena itu, dibutuhkan suatu aplikasi yang dapat membantu mempercepat penghitungan kompleksitas algoritma dalam memilih algoritma yang lebih efisien untuk membangun suatu aplikasi komputer.

1.2 Perumusan masalah

Berdasarkan latar belakang yang telah diuraikan, maka masalah yang diidentifikasi adalah bagaimana membuat aplikasi penghitung kompleksitas algoritma, yang dapat membantu mempercepat dalam pemilihan algoritma yang efisien untuk membangun suatu aplikasi komputer.

Adapun batasan masalah pembuatan aplikasi penghitung kompleksitas algoritma adalah:

1. Perhitungan kompleksitas algoritma meliputi perhitungan kompleksitas waktu dan kompleksitas ruang.
2. Perhitungan kompleksitas waktu algoritma dilakukan dengan cara menghitung jumlah baris yang dieksekusi.
3. Perhitungan kompleksitas ruang algoritma dilakukan dengan cara menghitung memori yang digunakan, yang dilihat dari type data yang digunakan.
4. Penghitungan kompleksitas hanya dilakukan terhadap satu prosedur atau satu fungsi algoritma.

5. Sintak dan logika algoritma yang diinputkan sudah dianggap benar.
6. Perhitungan kompleksitas algoritma mengacu pada parameter input.
7. Penghitungan kompleksitas hanya dilakukan terhadap algoritma dalam bahasa pemrograman Delphi.

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah :

1. Membuat aplikasi penghitung kompleksitas algoritma dari segi waktu dan ruang.
2. Menghitung kompleksitas waktu algoritma dengan menghitung banyaknya eksekusi baris yang dilakukan dalam suatu algoritma.
3. Menghitung kompleksitas ruang algoritma dengan menghitung memory yang dibutuhkan untuk menjalankan suatu algoritma.
4. Membantu perhitungan kompleksitas suatu algoritma untuk memilih algoritma yang efisien.

1.4 Metodologi penyelesaian masalah

Metode penyelesaian masalah dalam penelitian ini adalah sebagai berikut :

1. Studi Literatur
Mempelajari buku, makalah, dan referensi lain yang mendukung penyusunan tugas akhir. Studi literatur ini meliputi studi literatur tentang struktur data dan algoritma, studi literatur tentang kompleksitas algoritma, studi literatur tentang teknik kompilasi, dan studi literatur tentang bahasa pemrograman Borland Delphi 7.
2. Analisis dan Perancangan
Melakukan analisis dan perancangan implementasi aplikasi penghitung kompleksitas algoritma, dengan teori penghitungan kompleksitas algoritma dan teori kompilasi. Analisis dan perancangan perangkat lunak menggunakan tools Microsoft Visio 2003.
3. Implementasi dan Pengujian
Mengimplementasikan aplikasi perangkat lunak dengan menggunakan bahasa pemrograman Borland Delphi 7. Sedangkan pengujiannya dilakukan dengan menginputkan algoritma yang akan dihitung, serta dibandingkan kecepatan penghitungannya dengan penghitungan secara manual.
4. Pengambilan Kesimpulan
Pengambilan kesimpulan dilakukan berdasarkan analisis hasil pengujian.
5. Penyusunan Buku Laporan.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan pengujian, maka kesimpulan yang dapat diambil dari Tugas Akhir dengan judul Aplikasi Penghitung Kompleksitas Algoritma adalah :

1. Aplikasi Penghitung Kompleksitas Algoritma dapat menerima *input*, mendeteksi *variable* dan parameter *input*, menjalankan dan menghitung kompleksitas algoritma, serta menampilkan hasil penghitungan.
2. Grafik hasil penghitungan eksekusi baris dominan pada algoritma dengan jumlah masukan n yang iteratif, tidak semua mendekati teori kompleksitas algoritma asimptotik. Pada pengujian 1, yang kurang menunjukkan pendekatan terhadap teori kompleksitas asimptotik adalah algoritma nomor 3 dengan kompleksitas $O(n \log n)$.

5.2 Saran

Saran pengembangan Tugas Akhir dengan judul Aplikasi Penghitung Kompleksitas Algoritma adalah :

1. Penghitungan kompleksitas waktu dapat dikembangkan ke tahap penghitungan kompleksitas waktu asimptotik.
2. Penghitungan kompleksitas algoritma tidak hanya terbatas pada satu prosedur atau fungsi, tetapi dapat dilakukan terhadap lebih dari satu prosedur atau fungsi.
3. Sintak algoritma *input* yang dapat dieksekusi sistem dapat diperbanyak.
4. Penghitungan kompleksitas algoritma tidak hanya dilakukan terhadap algoritma dalam bahasa pemrograman Delphi, tetapi dapat dilakukan terhadap algoritma dalam bahasa yang lebih umum.
5. Diberikan saran pengganti penggunaan struktur data, apabila terdapat penggunaan *variable* yang membutuhkan memori yang cukup banyak.

Telkom
University

Daftar Pustaka

- [1] Alam, M. Agus J., 2001, *Belajar Sendiri Borlan Delphi 6.0*, Jakarta, PT Elex Media Komputindo Kelompok Gramedia.
- [2] *Handout Algoritma Pemrograman Dan Struktur Data 1*, Prodi Sistem Informasi UKDW.
<http://lecturer.ukdw.ac.id/anton/download/strukdat1.pdf>.
- [3] Hariyanto, Bambang, April 2004, *Teori Bahasa, Otomata, dan Komputasi serta Terapannya*, Bandung, Informatika Bandung.
- [4] Juarna, Asep, 2000, *Catatan Kuliah Teknik Kompilasi*.
<http://library.gunadarma.ac.id/files/disk1/1/jbptgunadarma-gdl-course-2000-asepjuarna-13-teknikko-i.pdf>
- [5] Levitin, Anany, *Introduction to the Design and Analisis of Algorithm*, Addison Wesley.
- [6] Munir, Renaldi, Ir., M.T., 2004, *Bahan Kuliah IF2251 Strategi Algoritmik*, Bandung, Departemen Teknik Informatika Institut Teknologi Bandung.
- [7] R., M. Arief Faizal, 2005, *Tugas Akhir - Translator PL/SQL pada Oracle ke Transact-SQL pada Microsoft SQL Server*, Bandung, STT Telkom.
- [8] Sedyono, Eko, 2005, *Teknik Kompilasi Teori dan Praktik*, Yogyakarta, ANDI OFFSET.
- [9] Sipser, Michael, 1997, *Introduction to The Theory of Computation*, PWS, Publishing Company
- [10] Soetanto, Harry, 2004, *Teknik Kompilasi*, Fakultas Teknologi Informasi Universitas Budi Luhur.
http://www.bl.ac.id/dosen/hari/Kuliah/buku_tekom.pdf.
- [11] Utdirartatmo, Furrar, 2001, *Teori Bahasa dan Otomata*, Yogyakarta, J&J Learning
- [12] Wahyudi, Bambang, 2004, *Pengantar Struktur Data dan Algoritma*, Yogyakarta, ANDI OFFSET.