

Journal of Engineering Science and Technology Management

| ISSN (Print) 2088-4842 | ISSN (Online) 2442-8795 |



Article

Aplikasi Algoritma Pengurutan Menggunakan Java BlueJ Sebagai Media Bantu Pembelajaran

Safni Marwa^{1,a}

¹ Prodi Teknik Informatika, Universitas Pahlawan

E-mail: a.safnimarwa@universitaspahlawan.ac.id

ARTICLE INFORMATION

Volume 2 Number 2

Received: 07 September 2022

Accepted: 27 September 2022

Publish Online: 29 September 2022

Online: at <https://JESTM.org/>

Keywords

Sorting Algorithm

Algorithm Animation

Java BlueJ

Programming Animation

ABSTRACT

Many research and project development has been done since a decade for algorithm animation. But as fundamental of the programming, sorting algorithm still difficult to understand, especially for the students who were novice or beginner in programming. To cope with that problem, one of the solutions was by creating tool for animation of sorting algorithm. As a teaching aid, this tool could help the students to understand the algorithm and at the end they could be able to build their own program. This research would be developed one of the sorting algorithms techniques that usually use by the lecturer: Merge sort. Java BlueJ will be the IDE (Integrated Development environment) to developed the algorithm animation. The Animation of sorting algorithm would provide the template featuring input random number which could be type by the user or through push the random number buttons, besides import existing data. The process would display visually movement of the bar structures and text where the output would show an ordered bar structure from small to large number. The purpose of the user interface modification was the user convenience to learn algorithm.

1. BACKGROUND

1.1 Introduction

Algoritma merupakan hal yang harus dipahami oleh seseorang yang mempelajari tentang pemrograman. Dalam pemrograman komputer, tuntutan akan pemahaman terhadap suatu masalah dan bagaimana logika berpikir secara terstruktur dalam menghadapi dan memecahkan masalah tersebut menjadi keterampilan yang penting untuk dikuasai. Sebagai alat pemecah masalah komputasi, algoritma memberikan penjelasan rinci tentang langkah – langkah program komputasi dalam menyiapkan bagaimana hubungan antara *input* dan *output* saling berinteraksi dalam proses kerjanya. Diantara contoh yang sangat umum adalah bagaimana mengurutkan serangkaian bilangan atau file yang acak.

Untuk menyelesaikan masalah ini, banyak cara atau metode yang dapat dilakukan. Mempelajari tentang algoritma seperti halnya algoritma pengurutan memerlukan pemahaman yang cukup akan sebuah konsep berpikir dalam hal ini logika. Untuk membantu dalam proses pembelajaran tentang algoritma pengurutan tersebut, dengan membaca referensi dan mendengarkan penjelasan tutorial saja, akan menjadi sulit. Sebagaimana yang disampaikan Younger & Narayanan (Marwa, 2009) dalam IEEE International Conference: *“Abstract procedural concepts, such as algorithm, are notoriously difficult for students to learn and difficult to teach”*.

Kecepatan dalam pemahaman materi algoritma pengurutan akan berbeda dan menjadi lebih baik apabila proses belajar dibantu dengan simulasi berupa aplikasi yang dapat memberikan gambaran visual tentang contoh permasalahan algoritma seperti halnya pengurutan. Karena sistem visual manusia diolah menggunakan ‘prosesor informasi’ yang canggih, maka mempelajari sesuatu dengan menggunakan alat bantu visual mungkin berdampak berbeda pada hasilnya. Oleh karena itu, salah satu cara untuk mengajarkan algoritma adalah dengan menggunakan bantuan visual, misalnya animasi.

Animasi algoritma akan memberikan visualisasi dari proses atau tahapan suatu algoritma. Apa yang terjadi pada data dan bagaimana penyelesaiannya secara terstruktur saat program dijalankan. Dengan demikian, pemahaman akan algoritma akan lebih mudah dengan menggunakan media animasi algoritma sebagai alat bantu dalam pembelajaran seperti algoritma pengurutan.

Sebagai salah satu algoritma dasar yang harus dikuasai, animasi algoritma pengurutan ini dapat dibangun menggunakan bahasa pemrograman apapun. Salah satu bahasa pemrograman yang sering dipelajari dan digunakan dalam pembuatan program adalah Java. Untuk penelitian ini digunakan bahasa pemrograman Java BlueJ dengan tampilan visual animasi seperti balok yang bergerak dengan text

proses serta hasil dari pengurutan yang dilakukan adalah angka. Adapun algoritma pengurutan yang divisualkan berupa algoritma penggabungan (*Merge Sort*).

1.2 Research Purposes

Aplikasi media bantu ini akan memberikan gambaran kepada mahasiswa khususnya tentang metode yang digunakan dalam penyelesaian permasalahan algoritma pengurutan *Merge Sort*. Sehingga dengan aplikasi berupa animasi sederhana, proses belajar algoritma pengurutan dapat berlangsung dengan lebih mudah, cepat dan mantap.

2. LITERATURE REVIEW

2.1 Animasi Algoritma

Dalam membuat sebuah program atau aplikasi komputer, dibutuhkan pemahaman akan kerangka berpikir yang terstruktur dalam pengambilan keputusan untuk pemecahan masalah. Agar dapat memahami dan membuat sebuah program, ada tiga hal yang harus menjadi perhatian yaitu: pertama sekali harus ada tugas atau permasalahan yang harus diselesaikan, kedua adalah metode atau proses pemecahan masalah (algoritma) dan ketiga adalah sebuah contoh implementasi atau penerapan algoritma tersebut kedalam bahasa pemrograman (Shaffer, 1998).

Animasi adalah salah satu cara yang efektif untuk memahami dan menunjukkan bagaimana algoritma bekerja. Kebutuhan untuk mengetahui dengan jelas terhadap sifat atau perilaku algoritma akan terjawab. Animasi dapat didefinisikan sebagai metode representasi dari sebuah proses atau serangkaian langkah – langkah melalui abstraksi dan logika yang menggunakan gambar, gerak dan teks. Ketika belajar sesuatu melalui gambar seperti objek grafis yang bergerak, maka sistem visual manusia akan menangkap, mengingat dan memahami dengan lebih baik dari pada metodologi belajar lainnya. Oleh karena itu, dalam rangka meningkatkan efektivitas proses pendidikan, maka penjelasan teksual yang didukung demonstrasi berupa animasi dapat membantu mahasiswa untuk terbiasa dengan algoritma (Yeh et al., 2006).

Untuk mempelajari algoritma pengurutan dengan menggunakan visualisasi merupakan peningkatan metode belajar dimana mahasiswa tidak hanya membayangkan algoritma saja tetapi mereka juga dapat membandingkan dan membuktikan dengan melihat proses yang ditampilkan oleh aplikasi animasi algoritma. Hampir semua animasi algoritma yang telah diciptakan menyediakan algoritma pengurutan. Itu menunjukkan bahwa algoritma pengurutan memiliki pengaruh besar pada pekerjaan komputasi.

2.2 Algoritma Pengurutan

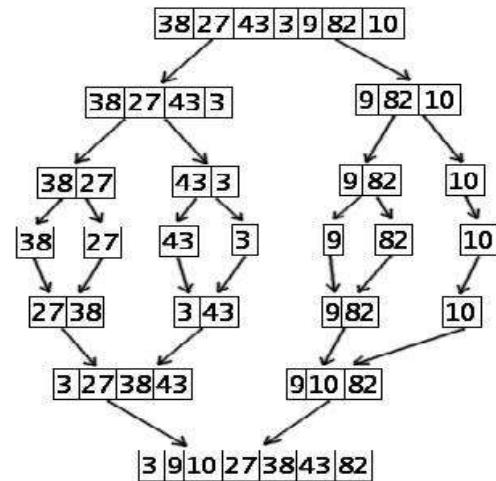
Algoritma pengurutan adalah salah satu algoritma komputasi dasar di mana implementasinya bergantung pada aplikasi. Algoritma ini biasanya digunakan untuk memfasilitasi proses pencarian dengan mengatur kembali serangkaian data atau objek dalam urutan tertentu. Sebagai manfaat, algoritma pengurutan ini membuat penyelesaian masalah menjadi lebih efisien, cepat dan mudah.

Ada dua kategori algoritma pengurutan sesuai dengan media penyimpanan dan pengolahan data. Pertama adalah pengurutan internal yang bekerja dengan array dan yang lainnya pengurutan eksternal dengan beberapa file yang diurutkan. Kedua kategori ini masih menggunakan algoritma dasar yang sama. Secara umum algoritma pengurutan dimulai dengan menemukan elemen terkecil, menukar, dan kemudian memeriksa untuk yang benar sebelum mengulangi proses kembali.

Ada banyak teknik untuk penyelesaian algoritma pengurutan, beberapa diantaranya dikembangkan berdasar yang telah ada. Sebagai contoh: *Shell sort* yang dikembangkan berdasarkan algoritma pengurutan *Insertion sort* (penyisipan), sementara itu *Heap sort* dikembangkan berdasarkan algoritma pengurutan *Selection sort*. Selain itu, juga ada teknik lainnya yang sering diberikan kepada mahasiswa untuk menjelaskan tentang algoritma pengurutan, seperti *Merge sort*, *Bubble sort*, dan *Quick sort* (Simmons, 2009).

2.3 Merge Sort Algorithm

Prinsip kerja dari algoritma pengurutan *Merge sort* adalah sejumlah data acak yang akan diurutkan dalam prosesnya dilakukan pembagian dengan jumlah data yang sama apabila genap ($N : N$) atau jumlah kelompok data kedua kurang satu dari kelompok data pertama apabila ganjil ($N : N-1$). Pembagian ini dilakukan sampai kelompok data terkecil yang kemudian akan dibandingkan satu sama lainnya untuk kemudian diurutkan. Setelah berurutan, maka selanjutnya data yang dibagi atas kelompok – kelompok tadi digabungkan kembali dan diurutkan sehingga diakhir akan diperoleh data tunggal yang berurutan (Koffman & Wolfgang, 2005).



Gambar 1. Merge Sort Algorithm

2.4 Bahasa Pemrograman Java

Ada berbagai jenis bahasa pemrograman telah ditemukan dan dipelajari sejak dua *decade* lalu dan banyak dari bahasa pemrograman ini telah digunakan untuk animasi algoritma. Untuk membuat sebuah aplikasi animasi digunakan bahasa pemrograman yang independen, yang berarti tidak ada pembatasan dalam menggunakan bahasa pemrograman apapun. Jumlah perbandingan dan pergerakan data sebagai analisis kinerja dapat dihitung dalam bahasa apapun. Diantara bahasa pemrograman yang telah digunakan untuk membangun animasi algoritma misalnya Pascal, C, C++, dan salah satunya yang telah digunakan untuk penelitian ini, bahasa pemrograman Java.

Bahasa pemrograman Java telah menjadi populer sejak tahun 1995 ketika beberapa jenis bahasa pemrograman Java dapat dimasukkan dalam halaman web. Program khusus ini disebut Applet, dan bahasa pemrograman Java yang datang dengan standar yang dibangun dengan kelengkapan grafis, membuat keberadaan bahasa pemrograman ini menjadi lebih kuat. Bahasa pemrograman Java diciptakan sebagai alternatif terhadap Bahasa pemrograman C++ dan mereka berbagi dari asal yang serupa.

Berikut beberapa karakteristik yang menjadi kekuatan dari bahasa pemrograman Java (Pietkiewicz – Koutny, 2009):

- a. *Program Portability*
- b. *Simple*
- c. *Strongly Typed*
- d. *Garbage Collection*
- e. *Completely Object Oriented*
- f. *Arrays handles as object*
- g. *Multithreaded*
- h. Kaya akan *library* yang standar

Untuk penelitian ini, aplikasi dibangun dan dikembangkan menggunakan aplikasi Java tepatnya BlueJ sebagai lingkungannya (*Integrated*

Development Environment/IDE). Mengambil keuntungan dari bahasa pemrograman Java, aplikasi ini juga dapat dibangun dan diterapkan di lingkungan lain seperti misalnya NetBeans dan Eclipse.

2.5 Penelitian Terkait

Ada banyak animasi algoritma yang telah dibuat dan digunakan sebagai media bantu mengajar. Sebagai contoh adalah Balsa (1984), Tango (1990), Polka (1993) dan Animal (2002) untuk sistem animasi algoritma (Zhou et al., 2008). Di internet/*World Wide Web* (WWW), tersedia banyak animasi algoritma dengan fungsi tertentu seperti penyortiran algoritma, pemrograman linier dan pohon biner terurut.

Algoritma animasi telah terkenal sejak 1981 dimana film *Sorting Out Sorting* berhasil digunakan untuk media dalam pengajaran mahasiswa ilmu komputasi. Di film ini, Baecker dan kawan - kawan menunjukkan sembilan algoritma pengurutan yang berbeda (Baecker et al., 1997).

Diikuti oleh Balsa di Brown University dimana sistem ini menunjukkan bahwa animasi algoritma dapat berinteraksi dengan berbagai pengguna. Setelah itu, Balsa II rilis pada tahun 1988 sejak 4 tahun dikembangkan yang memiliki kepustakaan animasi yang luas dan canggih. Pada tahun 1990, Stasko kemudian memperkenalkan Tango sebagai sistem animasi algoritma yang diimplementasikan berdasarkan *framework*.

Tango di buat untuk mengatasi masalah seperti perancangan animasi dan kesederhanaannya (Stasko, 1990). Setelah tiga tahun pengembangan, Stasko kembali menyampaikan animasi baru yang canggih bernama Polka. Polka berkembang dengan C++ sebagai bahasa pemrograman dan berjalan pada UNIX sebagai sistem operasi. Namun demikian, Polka masih berkembang dan sekarang memiliki sebuah versi yang memperbolehkan Polka bekerja berdasarkan sistem operasi Windows (GVU Center, 2002).

3. METHODOLOGY

Metodologi yang digunakan pada penelitian ini menggunakan langkah – langkah metode SDLC (*Software Development Life Cycle*) yang terbagi dalam 5 tahap, yaitu :

1. Perencanaan;
2. Pengidentifikasi masalah;
3. Analisa;
4. Perancangan dan Pembuatan Aplikasi; dan 5. Pengujian Aplikasi.

Setelah semua tahap ini dilalui, langkah terakhir adalah mendokumentasikan semua tahapan tersebut.

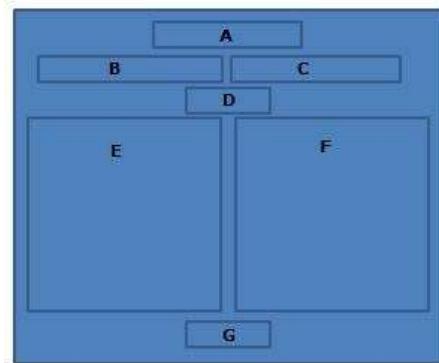
4. RESULTS AND DISCUSSION

Berdasarkan permasalahan akan sulitnya pemahaman tentang algoritma dimana algoritma pengurutan adalah materi yang mendasar bagi mahasiswa, maka perlu sekali adanya sebuah media bantu pembelajaran yang melibatkan unsur penglihatan/visual. Secara spesifik adalah bagaimana membuat sebuah aplikasi dalam memahami materi algoritma pengurutan dengan menggunakan animasi yang berupa pergerakan sebuah struktur dengan menggunakan bahasa pemrograman Java.

Tahap perancangan dilakukan dengan menggunakan UML (*Unified Modelling Language*) dalam hal ini diagram use case, dan diagram aktivitas untuk perancangan proses. Sedangkan untuk perancangan *template* dari *user interface* digunakan Microsoft Visio.

4.1. User Interface

Tampilan pengguna (*User Interface*) yang sederhana dari tool ini dirancang dengan fitur – fitur untuk memberikan hasil yang diharapkan sebagaimana terlihat di gambar 2.

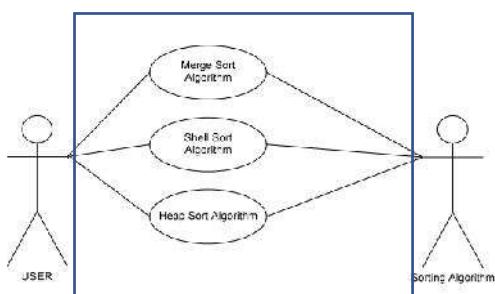


Gambar 2. Rancangan *User Interface*

Keterangan:

- A. Tombol *Combo Box* : memberikan pilihan untuk melakukan metode pengurutan *Merge sort*;
- B. Tombol *Random Number* : memberikan sejumlah bilangan acak;
- C. Tampilan bilangan acak;
- D. Tombol *Run* : mengaktifkan tampilan algoritma dengan teks dan struktur batang;
- E. Tampilan teks;
- F. Tampilan struktur batang; dan
- G. Tombol *Help* : memberikan informasi tentang metoda algoritma yang dipilih.

4.2. Use Case Diagram

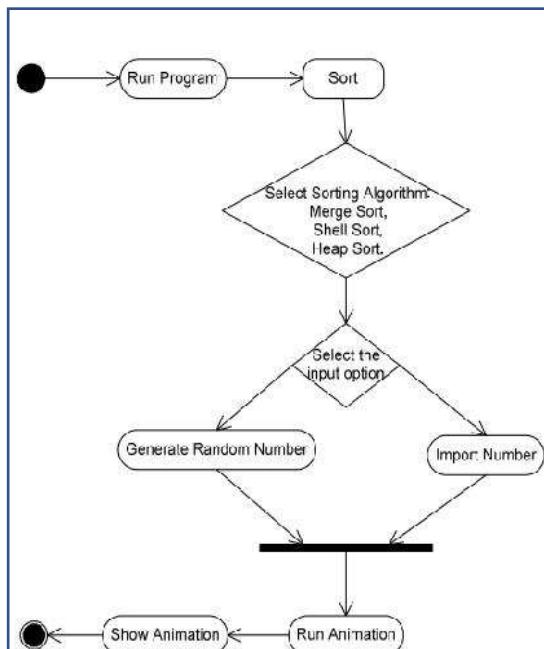


Gambar 3. *Use Case Diagram*

Pada *use case diagram*, aktor terdiri dari *user* dan sistem aplikasi algoritma pengurutan. Untuk sistem menampilkan *use case* dari pilihan untuk menjalankan algoritma *Merge sort*.

4.3. Activity Diagram

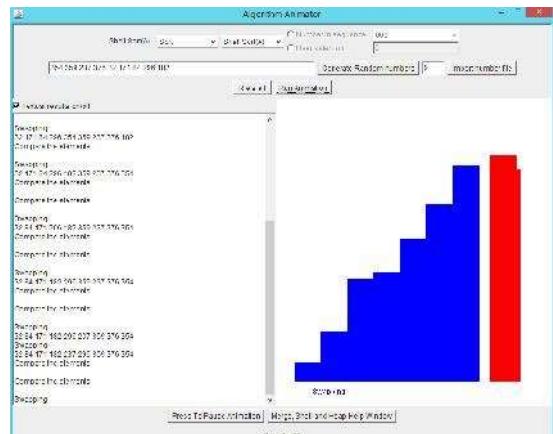
Activity diagram memberikan gambaran akan urutan proses atau menunjukkan langkah – langkah dari kegiatan yang dilakukan dalam menjalankan aplikasi animasi algoritma pengurutan tersebut.



Gambar 3. *Activity Diagram*

4.4. Implementasi

4.4. Implementasi



Gambar 4. Hasil Implementasi *User Interface*

5. CONCLUSION

Kesimpulan dari penelitian ini adalah:

- a) Aplikasi dibuat dengan beberapa penambahan fitur dengan fungsi utama yang sudah diselesaikan adalah untuk menampilkan animasi dari algoritma pengurutan *merge sort*.
- b) b. Bilangan dapat di-kan dengan beberapa cara yaitu dengan menekan tombol *generate random number*, *input* dengan langsung mengetikan sejumlah bilangan acak (sesuai dengan batas jumlah bilangan yang diberikan) dan juga dapat dilakukan dengan meng-*import* dari *file* data yang tersimpan/ yang ada.
- c) Animasi yang diberikan berupa batang sehingga terlihat seperti grafik dengan variasi warna saat langkah – langkah pengurutan dilakukan (membandingkan, memeriksa dan memindahkan bilangan).

REFERENCES

Baecker, R., DiGiano, C. & Marcus, A. (1997). Software Visualization for Debugging Communication of The ACM_Vol. 40_ No. 4.

Cormen, Thomas H. et al. (2009). Introduction To Algorithms. 3rd Edition. USA: MIT Press.

Goodrich, M. T. & R. Tamassia. (2004). Data Structures and Algorithms in Java. Chichester: Wiley.

Koffman, E. B. & P. A. T. Wolfgang. (2005). Objects, Abstraction, Data Structures and Design Using Java. New York; Chichester: Wiley.

Marwa, Safni. (2009). Animation of Sorting Algorithm. Dissertation. Newcastle upon Tyne: Newcastle University.

Pietkiewicz – Koutny, M. (2009). Lecture Note: Advance Programming. Newcastle upon Tyne: Newcastle University.

Pranata, Anthony. (2005). Algoritma & Pemrograman. Yogyakarta: Graha Ilmu.

Shaffer, C. A. (1998). A Practical Introduction to Data Structures and Algorithm Analysis. New Jersey: Prentice Hall.

Simmons, M. (2009). Animating, Sorting & Searching Algorithms. CS – Newcastle University

Stasko, J. T. (1990). Tango: A Framework and System for Algorithm Animation. IEEE Computer Volume 23, Issue 9.

The Graphic, Visual and Usability (GVU) Center. (07 Dec 2011). Algorithm Animation. Retrieved 25th April 2014 from World Wide Web http://www.cc.gatech.edu/gvu/softviz/alg_oanim/algoanim.html

Wajhillah, Rusda. et al. (2011). Visualisasi Logika Algoritma Pengurutan Data Menggunakan Java. SNIT Proceeding'11.

Yeh, C. L., Greyling, J. H. & Cilliers, C. B. (2006). A Framework Proposal for Algorithm Animation Systems. Proceedings of SAICSIT 2006.

Zhou, X., Li, W., Xian, H., Lai, T., & Liang, H. (2008). Towards Context Modelling for Algorithm Animation. Proceedings of COMPSAC'08.