



Implementasi Struktur Data Antrian Queue dalam Sistem Penjadwalan Proses pada Sistem Operasi

Azura Trijayanti^{1*}, Intan Aulia², Nazwa Khairunisa³, Farhan Asyrof Hamadi Purba⁴,
Indra Gunawan⁵

¹⁻⁵Program Studi Teknik Informatika, Sekolah Tinggi Ilmu Komputer Tunas Bangsa
Pematangsiantar, Indonesia

E-mail: azuratrijayanti94@gmail.com¹, intanaulia829@gmail.com², nazwakhairunnisa654@gmail.com³,
farhanpurba65@gmail.com⁴, indra@amiktunasbangsa.ac.id⁵

Alamat: Jl.Jend Sudirman Blok A No.1-3 Pematangsintar

*Korespondensi penulis: azuratrijayanti94@gmail.com

Abstract. Data structures are a fundamental concept in computer science used to store, organize, and manage data within a program or computer system. The selection of the appropriate data structure plays a crucial role in optimizing both time and space efficiency in data processing. Generally, data structures are classified into two main categories: linear and non-linear structures. Linear data structures include arrays, linked lists, stacks, and queues, where elements are arranged in a sequential order. Non-linear data structures, on the other hand, include trees and graphs, which organize elements in more complex relationships. Each type of data structure has specific characteristics and operations that can impact the performance of applications, such as searching, insertion, and deletion of data. The choice of the right data structure is essential in algorithm design as it can significantly enhance the efficiency and effectiveness of a system or application.

Keywords: Data Structure, Operating System, Queue.

Abstrak. Struktur data merupakan konsep dasar dalam ilmu komputer yang digunakan untuk menyimpan, mengorganisir, dan mengelola data dalam suatu program atau sistem komputer. Pemilihan struktur data yang tepat sangat berpengaruh terhadap efisiensi waktu dan ruang dalam pemrosesan data. Secara umum, struktur data linear mencakup array, linked, stack, dan queue, dimana elemen-elemen disusun dalam urutan tertentu. Sementara itu, struktur data non-linear meliputi tree dan graph, yang menyusun elemen-elemen dalam hubungan yang kompleks. Setiap jenis struktur data memiliki karakteristik dan operasi spesifik yang dapat mempengaruhi kinerja aplikasi, seperti pencarian, penyisipan, dan penghapusan data. Pemilihan struktur data yang tepat sangat penting dalam perancangan algoritma, karena dapat meningkatkan efisiensi dan efektivitas suatu sistem atau aplikasi.

Kata Kunci: Queue, Sistem Operasi, Struktur Data.

1. LATAR BELAKANG

Struktur data adalah salah satu fondasi utama dalam bidang ilmu komputer yang berkaitan erat dengan cara mengorganisir dan menyimpan data di dalam komputer. Tujuan utama dari penggunaan struktur data adalah untuk memungkinkan akses dan manipulasi data yang efisien. Tanpa pemilihan struktur data yang sesuai, program komputer dapat menjadi tidak efisien atau bahkan gagal memenuhi tujuannya secara optimal (Aho, Hopcroft, & Ullman, 1983).

Dalam praktik pemrograman sehari-hari, pengembang perangkat lunak kerap menghadapi berbagai permasalahan terkait pengelolaan data. Struktur data membantu mengatasi permasalahan tersebut dengan menyediakan metode penyimpanan dan pengambilan

data yang sesuai dengan kebutuhan algoritma. Misalnya, pencarian data dalam skala besar akan lebih efisien menggunakan struktur seperti hash table atau binary search tree dibandingkan dengan pencarian linier biasa (Cormen et al., 2009).

Struktur data terbagi menjadi dua kategori utama, yaitu struktur data linear dan non-linear. Struktur data linear, seperti array, linked list, stack, dan queue, menyusun data secara berurutan dan umumnya digunakan dalam konteks pengolahan data yang memerlukan urutan tertentu (Jusduenar, et al., 2024). Struktur ini sangat bermanfaat dalam implementasi antrian proses, manajemen memori, atau pemrosesan teks (Goodrich & Tamassia, 2014).

Sebaliknya, struktur data non-linear seperti tree dan graph menyusun data dalam hubungan hierarkis atau saling terhubung. Struktur ini sangat penting dalam implementasi sistem file komputer, pohon keputusan dalam kecerdasan buatan, serta algoritma pencarian rute pada aplikasi navigasi digital. Kemampuan untuk mengelola hubungan antar elemen data dengan efisien menjadikan struktur ini sangat vital dalam sistem kompleks (Knuth, 1997).

Pemahaman yang baik terhadap struktur data memungkinkan pengembang untuk merancang algoritma yang lebih efisien (Ginting et al., 2024). Misalnya, algoritma pencarian dan penyortiran (sorting) sangat bergantung pada struktur data yang digunakan. Binary search tree, heap, dan trie masing-masing menawarkan keuntungan yang berbeda dalam hal efisiensi waktu dan penggunaan memori (Weiss, 2011).

Lebih dari itu, struktur data juga berperan penting dalam pengembangan sistem berskala besar seperti basis data, sistem operasi, dan compiler. Dalam basis data, struktur data seperti B-tree dan hash index digunakan untuk mempercepat akses ke data. Sistem operasi menggunakan struktur seperti queue dan priority queue untuk penjadwalan proses, sedangkan compiler menggunakan tree untuk membentuk struktur sintaks (Aho et al., 2006).

Dalam konteks pendidikan, penguasaan struktur data menjadi materi wajib dalam kurikulum ilmu komputer dan rekayasa perangkat lunak. Mahasiswa diharapkan tidak hanya memahami teori di balik struktur data, tetapi juga mampu mengimplementasikannya dalam berbagai bahasa pemrograman. Hal ini karena struktur data tidak hanya sebagai alat bantu, tetapi juga sebagai penentu kualitas dan efisiensi dari solusi algoritmik yang dikembangkan (Shaffer, 2013).

Selain itu, pemilihan struktur data yang tepat berpengaruh langsung terhadap kompleksitas algoritma. Kesalahan dalam memilih struktur data dapat mengakibatkan waktu eksekusi yang lambat atau penggunaan memori yang berlebihan. Oleh karena itu, dalam perancangan sistem yang skalabel dan responsif, pemahaman terhadap struktur data merupakan kompetensi yang tidak dapat diabaikan (Sedgewick & Wayne, 2011).

Sebagai, struktur data merupakan komponen fundamental dalam pengembangan sistem informasi dan perangkat lunak modern. Penggunaan struktur data yang tepat dapat mengoptimalkan efisiensi waktu dan ruang dalam penyimpanan serta pemrosesan data. Oleh karena itu, penguasaan terhadap konsep dan implementasi berbagai jenis struktur data menjadi keharusan bagi siapa pun yang ingin mendalami dunia teknologi informasi secara profesional (Tanenbaum, 1995).

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan deskriptif kualitatif dengan metode studi kepustakaan (library research). Pendekatan ini dipilih karena fokus utama penelitian adalah mengkaji dan menganalisis konsep, karakteristik, fungsi, serta implementasi struktur data queue melalui berbagai sumber literatur ilmiah dan dokumen teknis yang relevan. Pendekatan kualitatif memungkinkan penulis untuk menggali makna dan pemahaman mendalam mengenai peran struktur data queue dalam berbagai sistem komputasi modern.

Sumber data dalam penelitian ini terdiri atas buku teks ilmiah, artikel jurnal terindeks, modul perkuliahan, dan dokumentasi teknis yang memuat pembahasan terkait struktur data, khususnya struktur data antrian (queue). Beberapa referensi utama diambil dari literatur klasik dan kontemporer dalam ilmu komputer, seperti karya Cormen et al. (2009), Weiss (2011), Goodrich & Tamassia (2014), serta Tanenbaum (2015).

Teknik pengumpulan data dilakukan melalui dokumentasi dan telaah literatur. Penulis mengidentifikasi dan mengumpulkan berbagai materi yang relevan untuk dianalisis secara sistematis. Fokus analisis diarahkan pada aspek-aspek penting seperti prinsip kerja queue (FIFO), operasi dasar (enqueue dan dequeue), jenis-jenis queue (linear, circular, priority, dan deque), kelebihan dan kekurangan, serta implementasi dalam berbagai konteks pemrograman dan sistem operasi.

Analisis data dilakukan secara kualitatif-interpretatif, yaitu dengan mengorganisasi data hasil telaah literatur, mengklasifikasikannya berdasarkan tema, kemudian mendeskripsikan secara logis dan sistematis dalam bentuk narasi ilmiah. Penulis juga melakukan perbandingan antar sumber untuk menemukan kesamaan dan perbedaan dalam pemahaman konsep serta praktik implementasi struktur data queue.

3. HASIL DAN PEMBAHASAN

Queue atau antrian adalah salah satu jenis struktur data linear yang menerapkan prinsip FIFO (First In First Out), di mana elemen yang pertama kali masuk akan menjadi elemen pertama yang keluar. Konsep ini mencerminkan banyak kejadian dalam kehidupan nyata, seperti antrian pada loket pelayanan publik, antrian cetak pada printer, dan antrian data dalam transmisi jaringan komputer (Aulia, 2023).

Struktur data queue memiliki sejumlah kelebihan. Pertama, queue sangat berguna dalam penjadwalan tugas dan manajemen antrian pada sistem komputer. Dalam sistem operasi, struktur ini digunakan untuk mengatur proses berdasarkan urutan kedatangan. Kedua, queue memudahkan pengaturan eksekusi tugas-tugas yang datang secara berurutan, seperti dalam layanan pelanggan berbasis giliran (Latifa et al., 2024).

Meskipun demikian, queue juga memiliki beberapa keterbatasan. Salah satunya adalah akses terhadap elemen-elemen di tengah antrian tidak dapat dilakukan secara langsung. Elemen hanya dapat ditambahkan dari belakang (rear) dan dihapus dari depan (front). Selain itu, ukuran antrian bisa jadi terbatas oleh kapasitas memori, terutama bila menggunakan array statis.

Terdapat dua operasi utama yang dapat dilakukan pada queue, yaitu enqueue dan dequeue. Enqueue adalah operasi untuk menambahkan elemen ke bagian belakang antrian, sementara dequeue adalah operasi untuk menghapus elemen dari bagian depan. Operasi-operasi ini berjalan secara bergantian sesuai dengan prinsip FIFO, memastikan bahwa data diproses dalam urutan kedatangannya.

Struktur queue terdiri dari dua pointer utama, yaitu front dan rear. Pointer front menunjukkan elemen pertama dalam antrian, sedangkan rear menunjukkan elemen terakhir yang ditambahkan. Elemen-elemen yang berada di antara front dan rear merupakan elemen aktif dalam antrian. Saat operasi dequeue dilakukan, elemen di posisi front akan dihapus dan pointer front bergerak maju. Sebaliknya, saat enqueue dilakukan, pointer rear akan berpindah ke posisi berikutnya (Goodrich & Tamassia, 2014).

Queue memiliki berbagai fungsi penting dalam sistem komputer modern. Beberapa di antaranya adalah: (1) sistem antrian layanan konsumen, (2) penjadwalan proses dalam sistem operasi, (3) pengelolaan memori, (4) algoritma Breadth-First Search pada graf, (5) manajemen antrian URL pada web crawler, (6) pengalokasian sumber daya seperti bandwidth dan memori, (7) pengelolaan buffer input/output, (8) pengaturan aliran data dalam jaringan komputer, (9) sistem pencadangan data (backup), dan (10) pengelolaan multithreading dan concurrency dalam sistem paralel. Queue juga digunakan dalam berbagai aplikasi lainnya, seperti manajemen event dan konfigurasi sistem (Shaffer, 2013).

Berdasarkan bentuk implementasinya, terdapat beberapa jenis queue. Linear queue (simple queue) adalah jenis paling dasar, di mana elemen ditambahkan dari belakang dan dihapus dari depan dalam urutan linear. Circular queue adalah variasi dari linear queue, namun pointer rear dapat kembali ke awal array jika ada ruang kosong, sehingga penggunaan memori lebih efisien. Berdasarkan penggunaannya, terdapat juga priority queue, yaitu struktur queue di mana setiap elemen memiliki prioritas tertentu dan elemen dengan prioritas tertinggi akan dilayani lebih dahulu. Selain itu, ada double-ended queue (deque), yang memungkinkan penambahan dan penghapusan elemen dari kedua ujung antrian (Weiss, 2011).

Queue dapat diimplementasikan menggunakan dua metode utama, yaitu menggunakan array dan menggunakan linked list. Pada implementasi array, digunakan dua pointer—front dan rear—untuk menunjukkan posisi elemen pertama dan terakhir. Kelebihan array adalah kecepatan akses indeks, tetapi memiliki kekurangan dalam fleksibilitas ukuran. Sementara itu, implementasi menggunakan linked list lebih fleksibel dalam hal ukuran, karena tidak tergantung pada ukuran tetap. Meskipun demikian, linked list membutuhkan memori lebih besar karena harus menyimpan pointer tambahan untuk setiap simpul (Cormen et al., 2009).

Beberapa contoh penggunaan queue dalam kehidupan nyata dan sistem komputer adalah sebagai berikut: dalam sistem operasi, queue digunakan untuk mengatur prioritas eksekusi proses; dalam sistem antrean pelayanan seperti bank atau rumah sakit; dalam sistem print untuk mengelola dokumen yang akan dicetak; dalam jaringan komputer untuk mengatur lalu lintas paket data; serta dalam sistem pemesanan tiket untuk mengatur urutan pengguna. Dengan banyaknya aplikasi tersebut, struktur data queue terbukti menjadi salah satu komponen penting dalam pengembangan sistem perangkat lunak modern (Tanenbaum, 2015).

4. KESIMPULAN DAN SARAN

Struktur data queue merupakan komponen penting dalam sistem operasi, terutama dalam penjadwalan proses. Queue bekerja dengan prinsip FIFO dan berfungsi untuk mengatur urutan eksekusi proses secara efisien dan adil. Terdapat berbagai jenis queue seperti linear, circular, priority, dan deque, yang masing-masing digunakan sesuai kebutuhan sistem, seperti dalam algoritma FCFS, Round Robin, dan Priority Scheduling.

Implementasi queue mempermudah manajemen proses, seperti dalam pengaturan antrian CPU, printer, dan buffer I/O. Meskipun memiliki banyak keuntungan seperti kesederhanaan dan efisiensi, queue juga memiliki keterbatasan, terutama dalam hal fleksibilitas dan pemanfaatan memori.

Secara keseluruhan, pemahaman dan penerapan struktur data queue sangat penting untuk membangun sistem operasi yang efisien, responsif, dan terstruktur.

DAFTAR REFERENSI

- Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1983). *Data structures and algorithms*. Addison-Wesley.
- Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, techniques, and tools* (2nd ed.). Pearson.
- Aulia, M. (2023). *Struktur data: Konsep, implementasi, dan aplikasi*. Deepublish.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). MIT Press.
- Ginting, S. H. N., Effendi, H., Kumar, S., Marsisno, W., Sitanggang, Y. R. U., Anwar, K., ... & Smrti, N. N. E. (2024). *Pengantar struktur data*. Penerbit Mifandi Mandiri Digital.
- Goodrich, M. T., & Tamassia, R. (2014). *Data structures and algorithms in Java* (6th ed.). Wiley.
- Knuth, D. E. (1997). *The art of computer programming, Volume 1: Fundamental algorithms* (3rd ed.). Addison-Wesley.
- Lafore, R. (2002). *Data structures and algorithms in Java* (2nd ed.). Sams Publishing.
- Langsam, Y., Augenstein, M. J., & Tenenbaum, A. M. (2006). *Data structures using C and C++* (2nd ed.). Prentice Hall.
- Malik, D. S. (2010). *Data structures using C++*. Cengage Learning.
- Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley.
- Shaffer, C. A. (2013). *Data structures and algorithm analysis in C++* (4th ed.). Pearson.
- Syamkalla, M. T., Khomsah, S., & Nur, Y. S. R. (2024). Implementasi algoritma CatBoost dan Shapley Additive Explanations (SHAP) dalam memprediksi popularitas game indie pada platform Steam. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 11(4), 777–786. <https://doi.org/10.25126/jtiik.1148503>
- Tanenbaum, A. S. (1995). *Structured computer organization* (3rd ed.). Prentice Hall.
- Weiss, M. A. (2011). *Data structures and algorithm analysis in Java* (3rd ed.). Pearson.