

TUGAS PROPOSAL
MATA KULIAH STRUKTUR DATA



OLEH:

Muslimin
(2022110059)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS INDO GLOBAL MANDIRI
PALEMBANG
2024

BAB I

PENDAHULUAN

Latar Belakang

Pemahaman mendalam tentang struktur data menjadi penting di era digital saat ini, tidak hanya bagi para profesional di bidang teknologi dan informasi saja, tetapi juga bagi individu di berbagai sektor lainnya. Struktur data menyediakan metode yang efisien untuk penyimpanan, pengaturan, dan pengelolaan data. Terdapat dua jenis struktur data yang sering digunakan, yaitu stack (tumpukan) dan queue (antrian).

Aplikasi dari konsep stack dan queue tidak hanya sebatas pada pemrograman dan pengembangan perangkat lunak, tetapi juga pemanfaatan kedua struktur data tersebut seringkali ditemukan dalam kehidupan sehari-hari. Seperti pada saat menerapkan stack dan queue untuk merancang solusi yang lebih efisien untuk masalah sehari-hari.

Terdapat tantangan dalam memahami dan menerapkan konsep stack dan queue, salah satu tantangan utamanya adalah memilih metode implementasi yang tepat, antara menggunakan array atau linked list, yang akan bergantung pada kebutuhan spesifik yang diinginkan. Selain itu, perlu juga memahami mengapa satu metode mungkin lebih cocok dibandingkan dengan metode lainnya.

Rumusan Masalah

1. Bagaimana stack dan queue dapat diterapkan dalam situasi kehidupan sehari-hari?
2. Metode implementasi apa yang paling sesuai untuk stack dan queue?
3. Apa alasan pemilihan metode tersebut?

Tujuan

1. Menjelaskan secara detail aplikasi dari stack dan queue dalam kehidupan sehari-hari.
2. Menjelaskan contoh penerapan dari stack dan queue menggunakan metode implementasi array dan linked list.
3. Menjelaskan alasan dari digunakannya array atau linked list dalam implemetasi stack dan queue.

BAB II

PEMBAHASAN

A. Konsep Dasar Stack dan Queue

Stack merupakan struktur data yang beroperasi dengan prinsip Last In First Out (LIFO), yaitu saat elemen terakhir yang dimasukkan ke dalam stack akan menjadi elemen pertama yang akan diambil. Prinsip ini sangat berguna dalam kasus-kasus seperti pemanggilan fungsi dalam pemrograman, dimana fungsi yang terakhir dipanggil harus diselesaikan terlebih dahulu.

Sementara itu, queue merupakan struktur data yang beroperasi dengan prinsip First In First Out (FIFO), yaitu saat elemen pertama yang dimasukkan ke dalam queue akan menjadi elemen pertama yang diambil. Queue banyak digunakan dalam pengaturan proses, penjadwalan, dan manajemen sumber daya dalam komputasi dan aplikasi nyata.

B. Aplikasi Stack dan Queue dalam Kehidupan Sehari-hari

Konsep stack dan queue tidak hanya terbatas pada bidang teknologi, tetapi juga memiliki aplikasi yang luas dalam kehidupan sehari-hari. Contoh stack dalam kehidupan sehari-hari adalah, buku yang terakhir ditumpuk adalah buku yang pertama diambil. Hal ini membantu dalam mengorganisasi dan mengakses item buku dengan lebih efisien. Contoh lainnya yakni pada saat menumpuk cucian piring, piring yang terakhir ditumpuk akan menjadi piring pertama yang diambil.

Di sisi lain, konsep queue hadir dalam sistem antrian, seperti antrian di bank atau di kasir. Sistem ini memastikan semua pelanggan dilayani berdasarkan urutan kedatangan mereka, menghindari kebingungan dan memastikan keadilan. Hal ini membuktikan bahwa prinsip FIFO dapat meningkatkan efisiensi dan keadilan dalam manajemen orang atau barang.

C. Implementasi Menggunakan Array dan Linked List

Implementasi stack dan queue dapat dilakukan melalui berbagai metode. Array dan linked list merupakan dua metode yang paling umum dalam mengimplementasikan keduanya. Array menawarkan keuntungan dalam hal waktu akses yang cepat karena elemen-elemen dapat diakses langsung melalui indeksnya. Namun, ukuran array yang statis bisa menjadi keterbatasan karena membutuhkan penentuan ukuran maksimal stack atau queue sejak awal.

Sementara itu, linked list memberikan fleksibilitas dalam hal ukuran dinamis. Elemen-elemennya dapat ditambahkan atau dihapus dengan mudah tanpa perlu khawatir tentang batasan ukuran. Hal ini menjadikan linked list pilihan yang baik untuk implementasi stack dan queue yang ukurannya berubah-ubah. Namun, waktu akses pada linked list bisa lebih lambat dibandingkan dengan array karena perlu melalui pointer untuk mengakses elemen tertentu.

D. Pemilihan Metode Implementasi

Pemilihan antara array dan linked list untuk implementasi stack dan queue bergantung pada beberapa faktor, termasuk ukuran data yang diinginkan dinamis atau statis, frekuensi operasi penambahan dan penghapusan data, serta kemudahan implementasi. Jika aplikasi membutuhkan ukuran stack dan queue yang tetap dan waktu akses yang cepat, array bisa menjadi pilihan yang lebih baik. Namun, jika aplikasi membutuhkan fleksibilitas dalam ukuran dan operasi penambahan serta penghapusan data yang sering, linked list dapat menjadi solusi yang lebih efektif.

Dalam memutuskan metode implementasi, penting untuk mempertimbangkan karakteristik aplikasi yang akan dikembangkan serta *trade-off* antara kecepatan akses dan fleksibilitas ukuran. Keputusan ini akan berdampak langsung pada efisiensi dan kepraktisan dari implementasi stack dan queue dalam aplikasi tersebut.

BAB III

PENUTUP

Kesimpulan

Stack dan queue, dengan prinsip operasionalnya yang berbeda, yakni Last In First Out (LIFO) dan First In First Out (FIFO), merupakan konsep fundamental dalam struktur data yang memiliki aplikasi luas baik dalam teknologi maupun dalam kehidupan sehari-hari. Stack, misalnya, dapat dilihat dalam tumpukan buku di meja kerja, dimana buku yang terakhir ditumpuk adalah yang pertama diambil saat dibutuhkan. Sementara itu, queue hadir dalam kehidupan kita sebagai sistem antrian di kasir supermarket, dimana pelanggan yang datang pertama dilayani pertama. Pemahaman mendalam tentang kedua konsep ini dan keputusan tepat antara menggunakan array atau linked list untuk implementasinya, tergantung pada kebutuhan aplikasi seperti ukuran data dan frekuensi operasi, esensial dalam merancang solusi yang efektif dan efisien, mengoptimalkan organisasi dan pemrosesan data dalam berbagai skenario.