

Halo semua! Selamat datang di presentasi video kami dengan judul "Pemecahan Masalah Antrian Rumah Sakit". Sebelum memulai perkenankan terlebih dahulu anggota kelompok saya

M. Naveed Sharaheel

M. Ridwan Ashari

Restu Wibisono

Indah Cahya Ramadhani

Syahreza Tidar W.L

Mari kita telusuri bagaimana penggunaan struktur data queue dapat membantu mengoptimalkan pengelolaan antrian di rumah sakit.

## SLIDE 1

**Mari kita mulai dengan pemahaman dasar. Apa itu queue?**

Queue adalah struktur data yang mengikuti prinsip "First-In-First-Out" (FIFO), di mana elemen yang pertama kali masuk ke dalam antrian akan menjadi yang pertama kali keluar. Operasi dasar pada struktur data queue melibatkan penambahan elemen ke dalam antrian (enqueue) dan penghapusan elemen dari antrian (dequeue). Antrian beroperasi mirip seperti antrian di kehidupan sehari-hari, di mana orang atau objek yang tiba lebih dulu akan dilayani lebih awal. Contoh penggunaan algoritma queue melibatkan skenario seperti manajemen antrian di rumah sakit, penjadwalan tugas dalam sistem operasi, dan berbagai sistem lain yang memerlukan pendekatan.

## SLIDE 2

**Namun, seperti halnya setiap sistem, penggunaan queue dalam manajemen antrian rumah sakit juga dapat menimbulkan beberapa permasalahan.**

1. \*Overflow dan Underflow:\*

- \*Overflow terjadi\* saat mencoba menambahkan elemen ke antrian yang sudah penuh.
- \*Underflow terjadi\* ketika mencoba menghapus elemen dari antrian yang kosong.

## 2. \*Penghapusan di Tengah Antrian:\*

- Jika perlu menghapus elemen di tengah antrian, dapat \*mengakibatkan penggeseran besar-besaran\*, meningkatkan kompleksitas waktu secara signifikan.

## 3. \*Masalah Sinkronisasi dalam Multithreading:\*

- Dalam aplikasi multithreading, operasi enqueue dan dequeue dari \*dua atau lebih thread\* dapat menyebabkan masalah sinkronisasi\*, memerlukan perhatian khusus untuk mencegah konflik.

# SLIDE 3

Bagaimana kita mengatasi permasalahan tersebut?

1. pertama, Atur batas maksimal antrian dan berikan penanganan khusus ketika mencoba menambahkan atau menghapus elemen ketika sudah mencapai batas tersebut.
2. Pkedua, ertimbangkan implementasi struktur data yang lebih fleksibel, seperti linked list, yang memungkinkan penghapusan elemen dengan kompleksitas waktu yang lebih rendah.
3. ketiga, Gunakan mekanisme sinkronisasi, seperti mutex atau semafor, untuk memastikan operasi enqueue dan dequeue dilakukan secara atomik.

# SLIDE 4

**Untuk mengakhiri presentasi ini, mari kita tinjau kembali.**

Queue, berdasarkan prinsip "First-In-First-Out" (FIFO), digunakan dalam banyak aplikasi, termasuk manajemen antrian di rumah sakit dan penjadwalan tugas dalam sistem operasi. Namun, beberapa permasalahan mungkin timbul, seperti overflow, underflow, kompleksitas waktu tinggi saat menghapus elemen di tengah antrian, dan masalah sinkronisasi dalam multithreading. Solusinya melibatkan pengaturan batas maksimal antrian, penggunaan struktur data yang lebih fleksibel seperti linked list, dan penerapan mekanisme sinkronisasi seperti mutex atau semafor. Dengan solusi ini, implementasi algoritma queue dapat dioptimalkan untuk menjawab tantangan khusus dalam berbagai konteks aplikatif.

**Terima kasih telah menyimak presentasi kami! Jika ada pertanyaan atau diskusi lebih lanjut, jangan ragu untuk mengajukan pertanyaan. Sampai jumpa di lain waktu!**