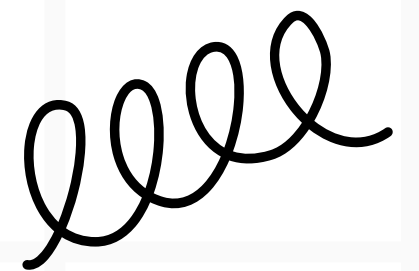
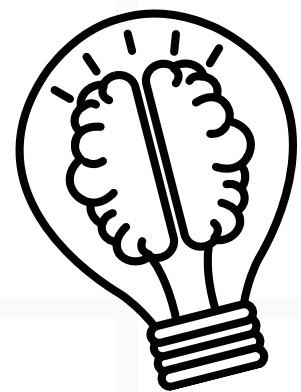
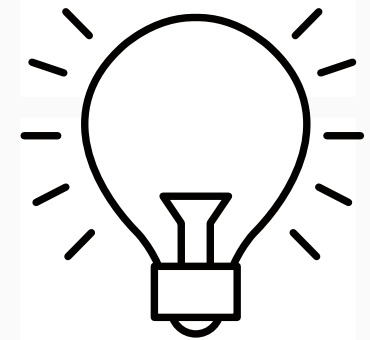


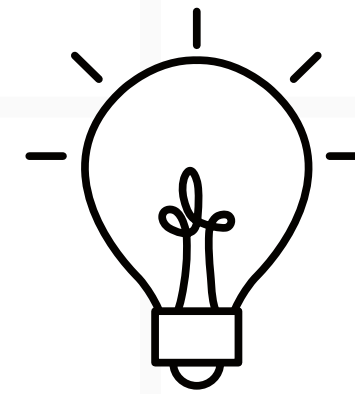
QUEUE

Oleh : Kelompok 4





TEAM



Hilmia Aqila

Kayla Elysia Sakina

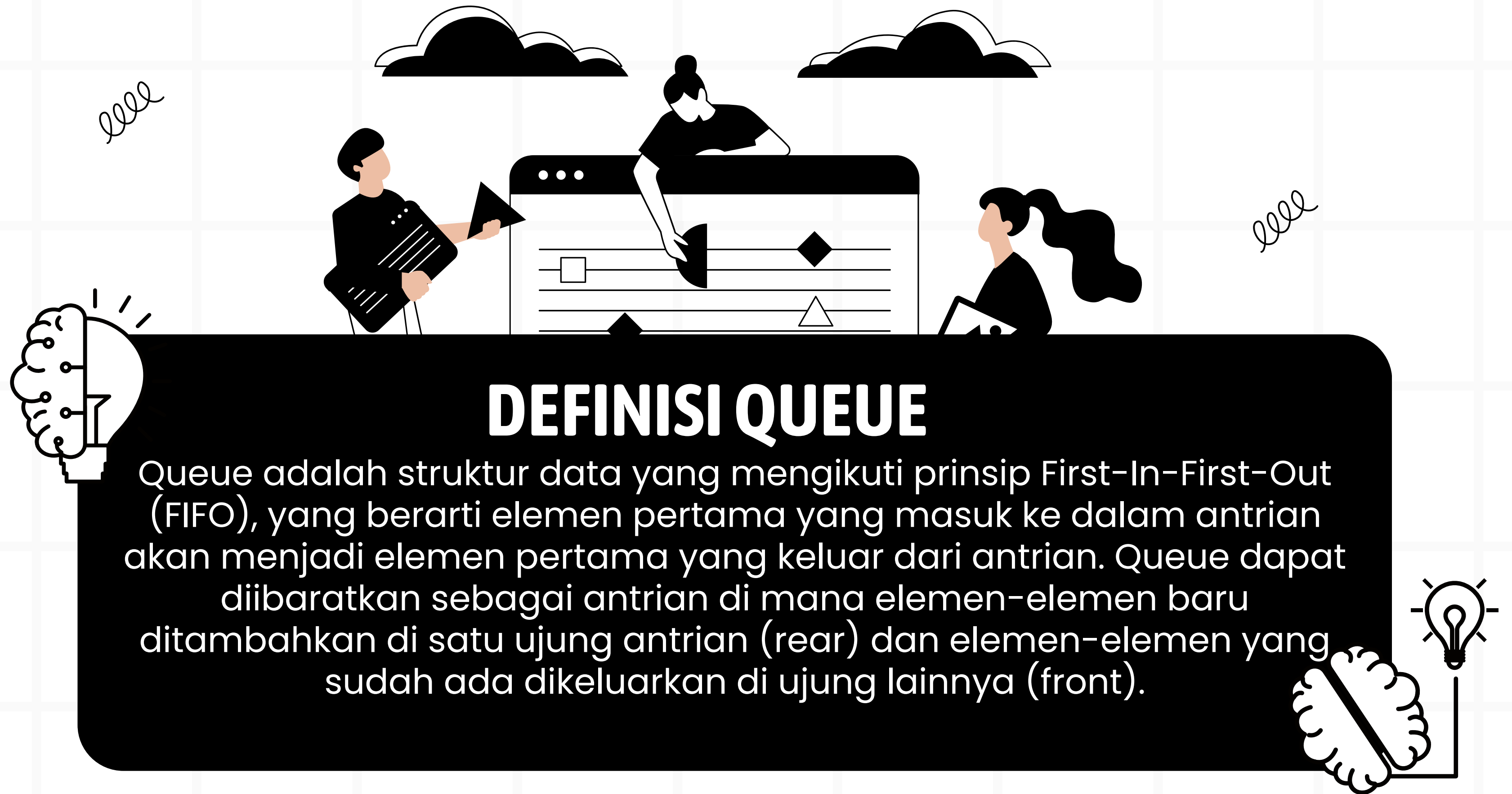
Deandra Adinda

Nilnarrusyda Aulia

Riska Nur Azizah

Sasa Izzatun Nisa



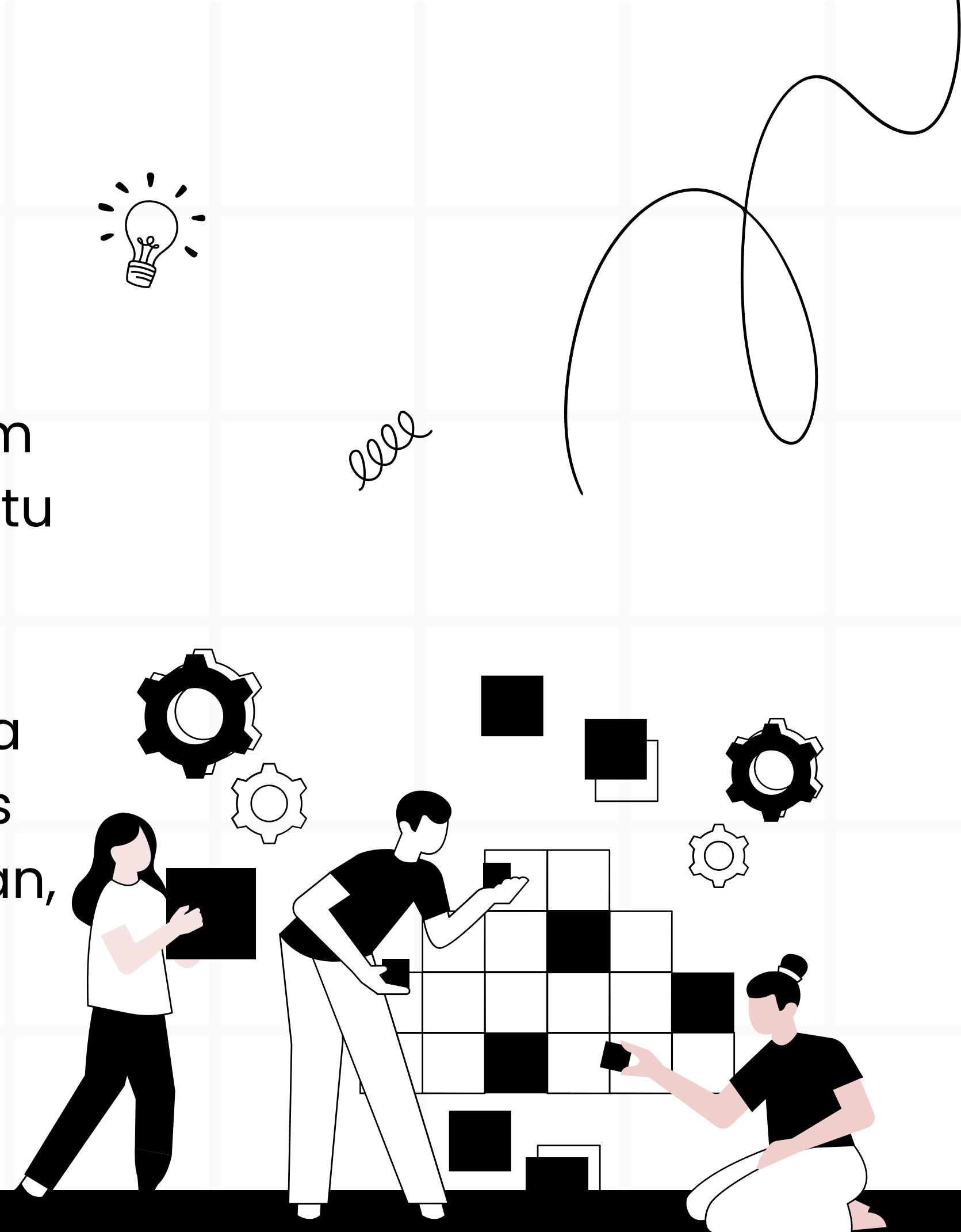


DEFINISI QUEUE

Queue adalah struktur data yang mengikuti prinsip First-In-First-Out (FIFO), yang berarti elemen pertama yang masuk ke dalam antrian akan menjadi elemen pertama yang keluar dari antrian. Queue dapat diibaratkan sebagai antrian di mana elemen-elemen baru ditambahkan di satu ujung antrian (rear) dan elemen-elemen yang sudah ada dikeluarkan di ujung lainnya (front).

FUNGSI QUEUE

Queue memiliki peran yang penting dalam berbagai aplikasi dan algoritma. Salah satu fungsi utamanya adalah mengatur dan mengelola antrean tugas atau operasi secara efisien. Dalam sistem komputasi, ia digunakan untuk menangani tugas-tugas seperti penjadwalan proses, antrean pesan, dan manajemen sumber daya.





JENIS-JENIS QUEUE

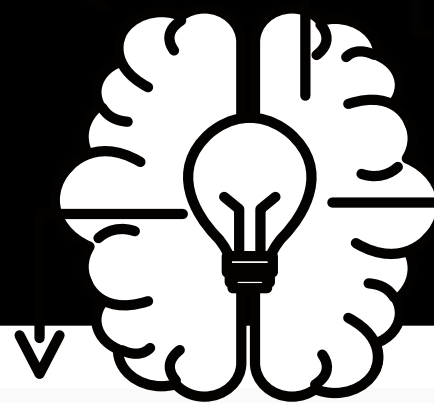
Secara umum ada 4 jenis struktur data queue, meliputi

- Simple Queue
- Circular Queue
- Priority Queue
- Double-Ended Queue (Deque)

llll

keuntungan

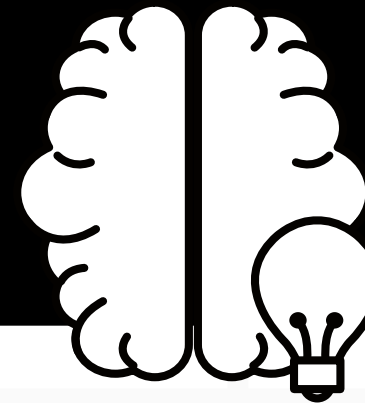
- Data berjumlah besar dapat dikelola dengan mudah dan efisien.
- Proses insert dan delete data dapat dilakukan dengan mudah karena mengikuti aturan first in first out (FIFO).
- Efisien dalam menangani tugas berdasarkan urutan kedatangan.



ANALISIS JAWABAN

keterbatasan

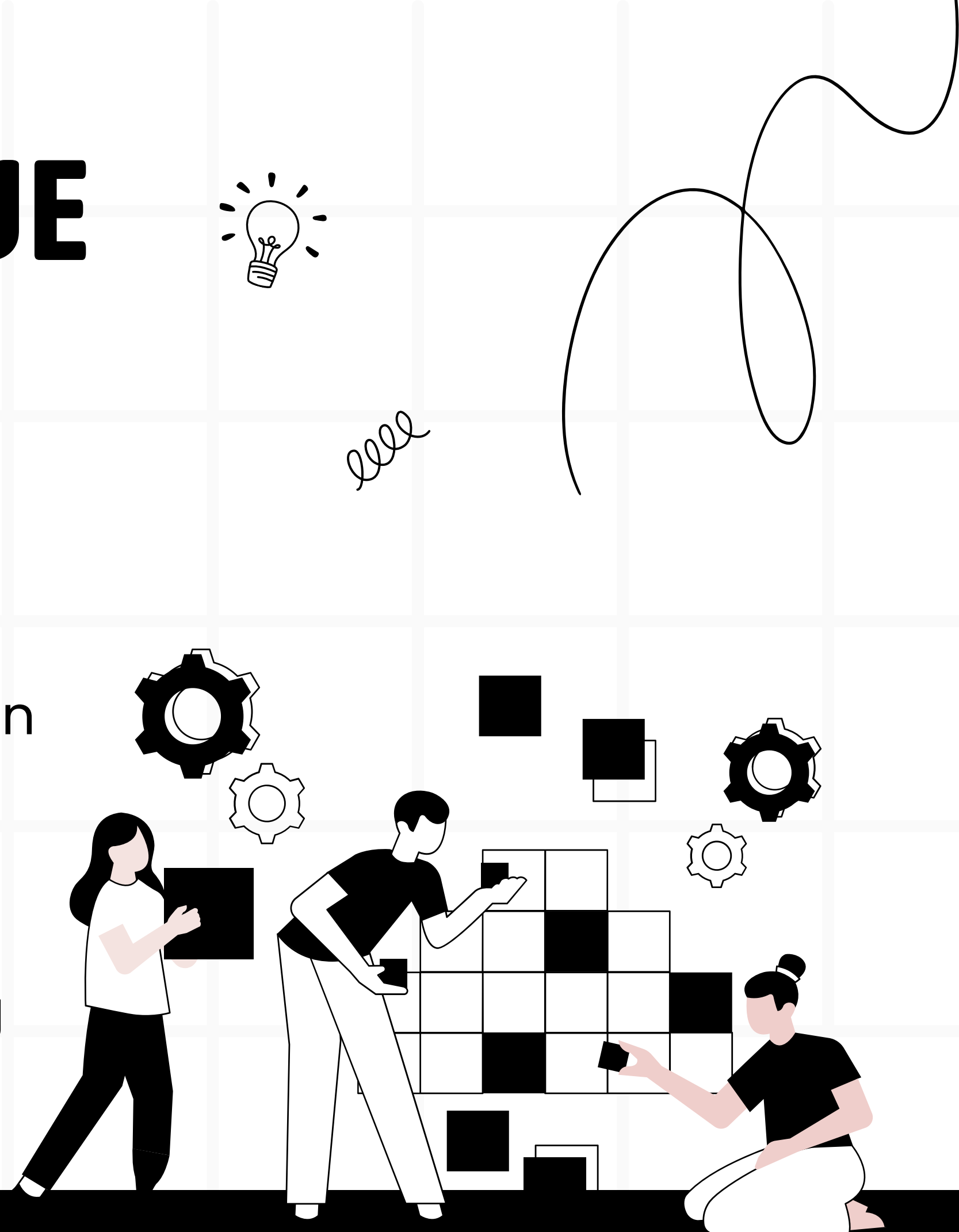
- Tidak efisien untuk pencarian elemen tertentu dalam antrean.
- Memerlukan alokasi memori yang cukup untuk menyimpan antrean.



KARAKTERISTIK QUEUE

Queue memiliki berbagai karakteristik sebagai berikut:

- Queue adalah struktur FIFO (First In First Out).
- Untuk menghapus elemen terakhir dari Queue, semua elemen yang dimasukkan sebelum elemen tersebut harus dihilangkan atau dihapus.
- Queue adalah daftar berurutan dari elemen-elemen dengan tipe data yang serupa.

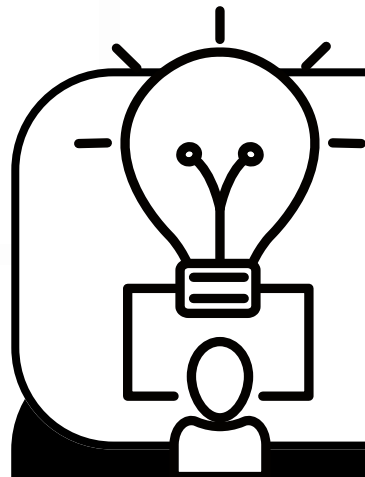
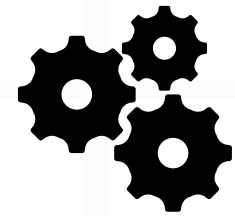


OPERASI DASAR QUEUE

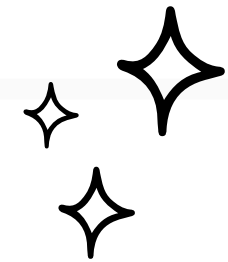
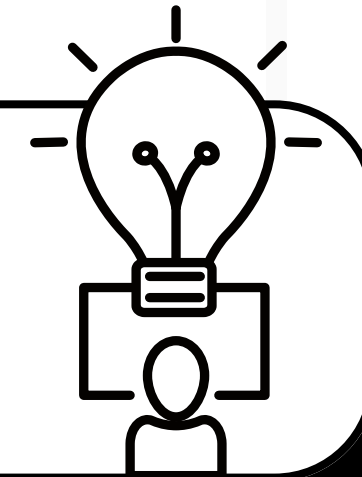
Queue adalah struktur data abstrak (ADT) yang memungkinkan operasi berikut:

- **Enqueue**: Menambahkan elemen ke akhir antrian
- **Dequeue**: Menghapus elemen dari depan antrian
- **IsEmpty**: Memeriksa apakah antrian kosong
- **IsFull**: Memeriksa apakah antrian sudah penuh
- **Peek**: Mendapatkan nilai bagian depan antrian tanpa menghapusnya
- **Initialize**: Membuat antrian baru tanpa elemen data (kosong)

KESIMPULAN

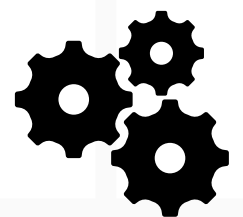
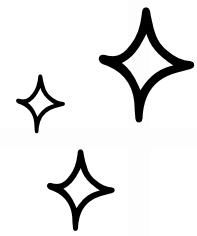


Tuliskan semua kesimpulan yang diperoleh dari voting dan diskusi.



Kita juga telah melihat contoh implementasi sederhananya, memperlihatkan bagaimana struktur data ini dapat digunakan dalam konteks antrean pelanggan pada kasir.

Dengan pemahaman ini, semoga kamu dapat mengimplementasikan konsep struktur data yang tepat dalam proyek-proyek aplikasi yang kamu kembangkan sehingga dapat meningkatkan efisiensi aplikasi dan juga menjadi seorang Software Engineer yang lebih andal.



TERIMA KASIH

Saya harap kamu bisa mendapatkan pengetahuan yang berguna dari presentasi ini. Semoga beruntung !

