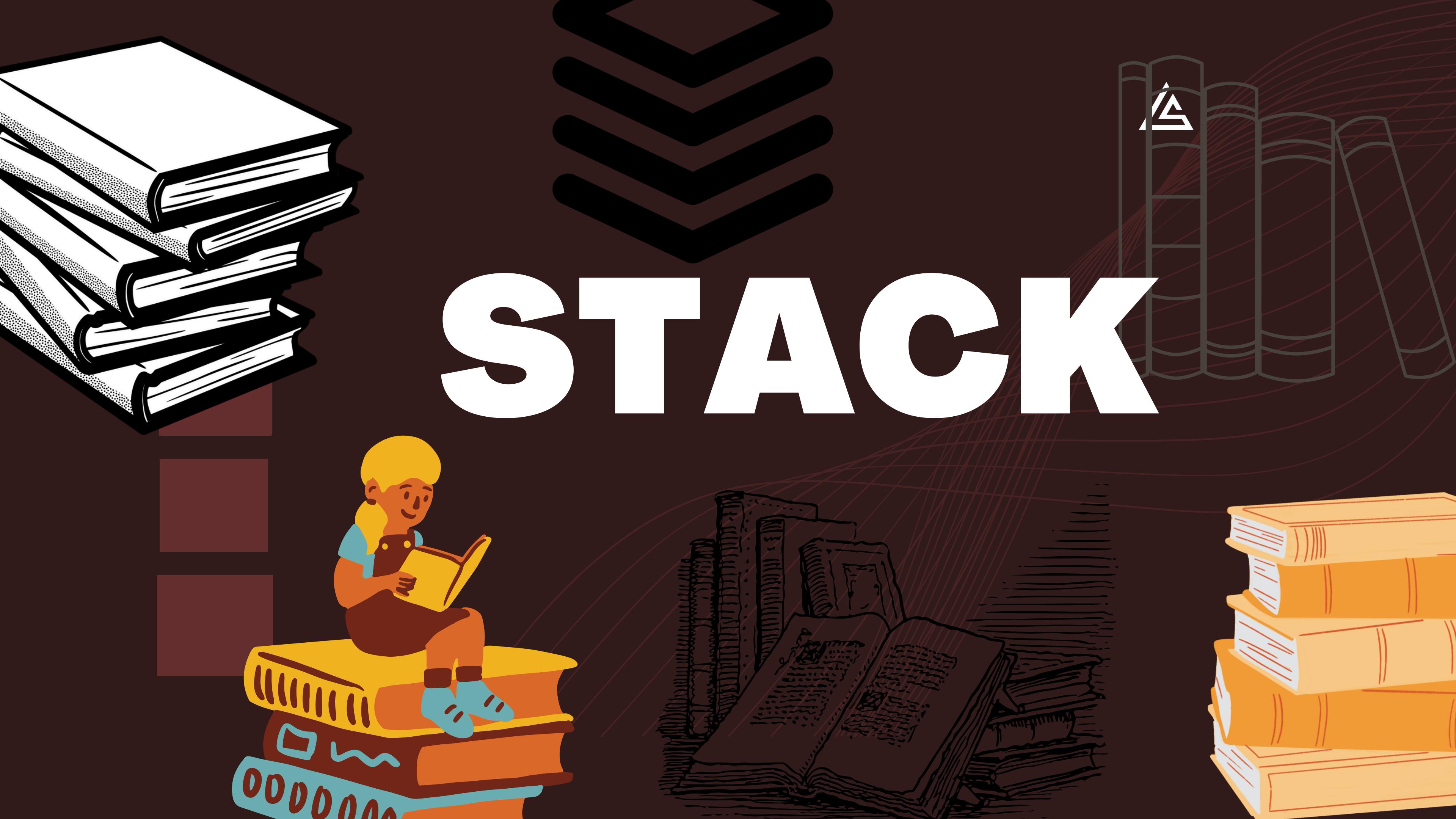


# STACK



# members of grup

**1.adhani aulia**

**4.dita wijayanti**

**2.akalily zahiroh**

**5.azzahra khansa**

**3.fitri arta**  
di yeratas.

**6.firyal adzhar**

**7.mumtaza fairuz**

# PENJELASAN

Stack adalah struktur data yang mengikuti prinsip Last In, First Out (LIFO), di mana elemen terakhir yang dimasukkan akan menjadi yang pertama dikeluarkan. Ini biasanya digunakan untuk melacak pemanggilan fungsi dalam pemrograman, mengevaluasi ekspresi aritmatika, manajemen memori, dan banyak lagi. Operasi utamanya adalah "push" untuk menambahkan elemen dan "pop" untuk mengeluarkan elemen teratas.



# FUNGSI DAN KEGUNAAN STACK

fungsi dan kegunaan struktur data stack adalah sebagai berikut:

- Struktur data stack digunakan dalam evaluasi dan konversi ekspresi aritmatika. Proses ini banyak dipakai untuk program kompiler.
- Stack digunakan dalam pemrograman rekursi.
- Digunakan untuk pemeriksaan tanda kurung.
- Stack digunakan dalam manajemen memori.
- Dipakai untuk memproses pemanggilan sebuah fungsi.

Salah satu contoh penerapan struktur data stack adalah fitur tombol back pada browser. Dimana browser akan menyimpan semua URL yang telah kita kunjungi sebelumnya dalam stack.

Setiap kali kita mengunjungi halaman baru, halaman itu ditambahkan di atas stack. Saat kita menekan tombol kembali, URL saat ini dihapus dari tumpukan, dan URL sebelumnya diakses.



# JENIS-JENIS STACK

1

Register stack merupakan stack yang hanya mampu menampung data dalam jumlah yang kecil. Kedalaman maksimum pada register stack cenderung dibatasi karena ukuran unit memorinya sangat kecil dibandingkan dengan memory stack.



2

Memory stack. Pada stack jenis ini, kedalaman dari stack cukup fleksibel dan mampu menangani dalam skala yang lebih besar dibandingkan jenis sebelumnya.

# KARAKTERISTIK STACK



- Stack digunakan pada banyak algoritma yang berbeda seperti Tower of Hanoi, Tree traversal, rekursi dll.
- Stack diimplementasikan dengan struktur data array atau linked list.
- Mengikuti prinsip operasi Last In First Out, yaitu elemen yang dimasukkan pertama akan muncul terakhir dan sebaliknya.
- Penyisipan dan penghapusan terjadi di satu ujung yaitu dari atas tumpukan.
- Apabila ruang memori yang dialokasikan untuk struktur data stack sudah penuh namun masih dilakukan operasi penyisipan elemen maka akan terjadi stack overflow.
- Apabila struktur data tidak memiliki elemen data atau kosong, namun tetap dilakukan operasi penghapusan maka akan terjadi stack underflow





# OPERASI-OPERASI DASAR PADA STACK



- **Push:** Menyisipkan elemen ke bagian atas stack
- **Pop:** Menghapus elemen atas dari stack
- **IsEmpty:** Memeriksa apakah stack kosong
- **IsFull:** Memeriksa apakah stack sudah penuh
- **Peek:** Mendapatkan nilai elemen teratas tanpa menghapusnya



# KELEBIHAN MENGGUNAKAN STACK



- **Manajemen data yang efisien:** Stack membantu mengelola data berdasarkan prinsip operasi LIFO yang tidak bisa dilakukan dengan linked list dan array.
- **Manajemen fungsi yang efisien:** Ketika suatu fungsi dipanggil, variabel lokal disimpan dalam stack, dan secara otomatis dihancurkan setelah dikembalikan.
- **Kontrol atas memori:** Stack memungkinkan kita untuk mengontrol bagaimana memori dialokasikan dan tidak dialokasikan.
- **Manajemen memori cerdas:** Stack secara otomatis membersihkan objek.
- **Tidak mudah rusak:** Stack tidak mudah rusak, oleh karena itu stack cenderung lebih aman dan dapat diandalkan.
- **Tidak mengizinkan pengubahan ukuran variabel:** Variabel pada stack tidak dapat diubah ukurannya.



# KEKURANGAN MENGGUNAKAN STACK



- **Ukuran memori terbatas:** Memori pada stack cukup terbatas.
- **Kemungkinan stack overflow:** Terlalu banyak membuat objek di stack dapat meningkatkan risiko stack overflow.
- **Akses acak tidak dimungkinkan:** Dalam stack, akses data secara acak tidak bisa dilakukan. Data yang dapat diakses adalah data yang berada pada elemen atas.
- **Dapat menyebabkan fungsi tidak terdefinisi:** Ketika penyimpanan variabel akan ditimpa, kadang-kadang akan menyebabkan perilaku fungsi atau program yang tidak terdefinisi.
- **Penghentian yang tidak diinginkan:** Jika stack berada di luar memori maka dapat menyebabkan penghentian yang tidak normal.





# TERIMA KASIH !!

