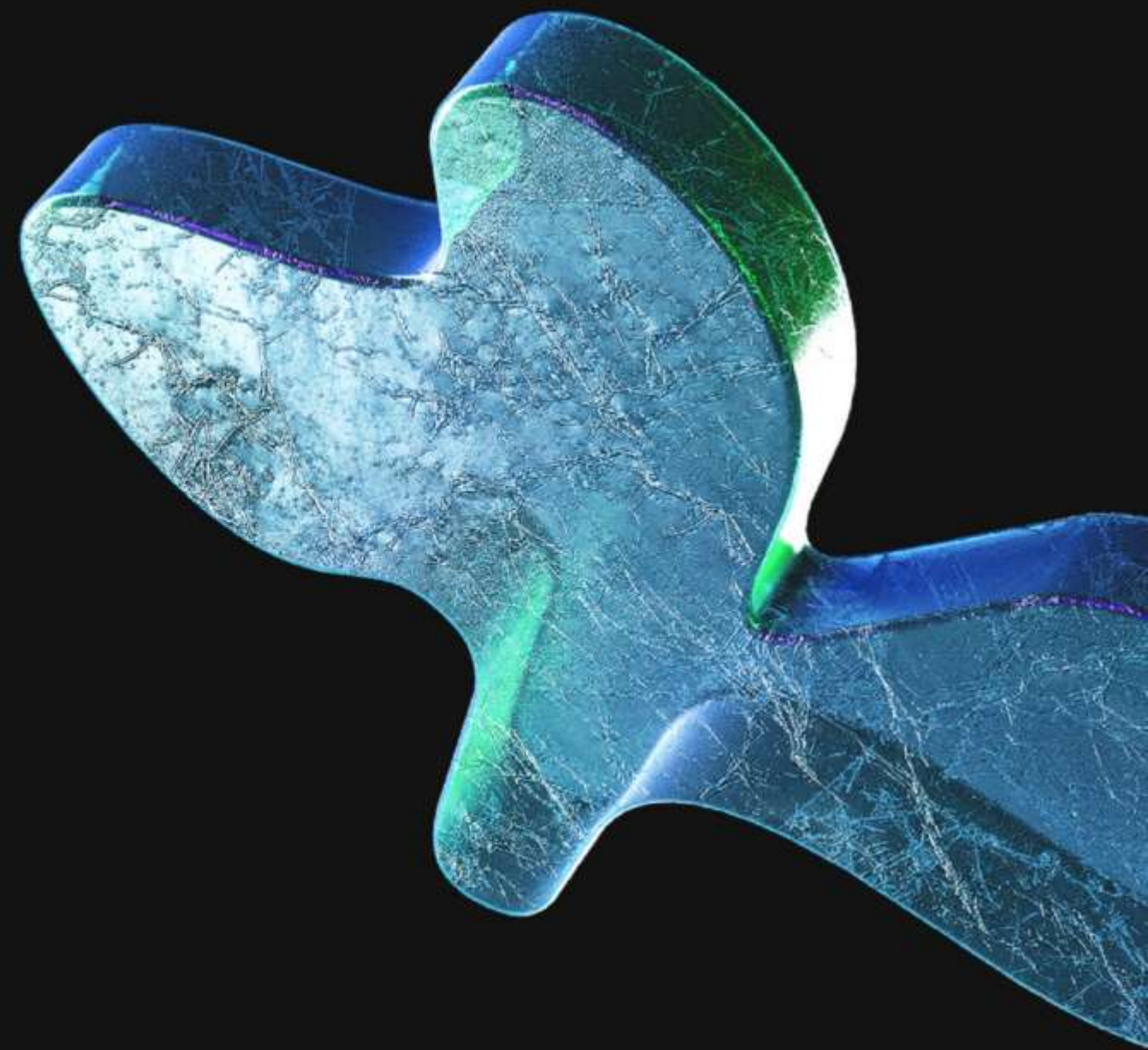




STACK DAN DOUBLE STACK



KELOMPOK 4



MUHAMMAD RAYYAN GHIFARI

SADIN YUSUF ARDIKA

DIMAS AUFA MUHAMMAD ZAKI

GALIH DWI NUGROHO

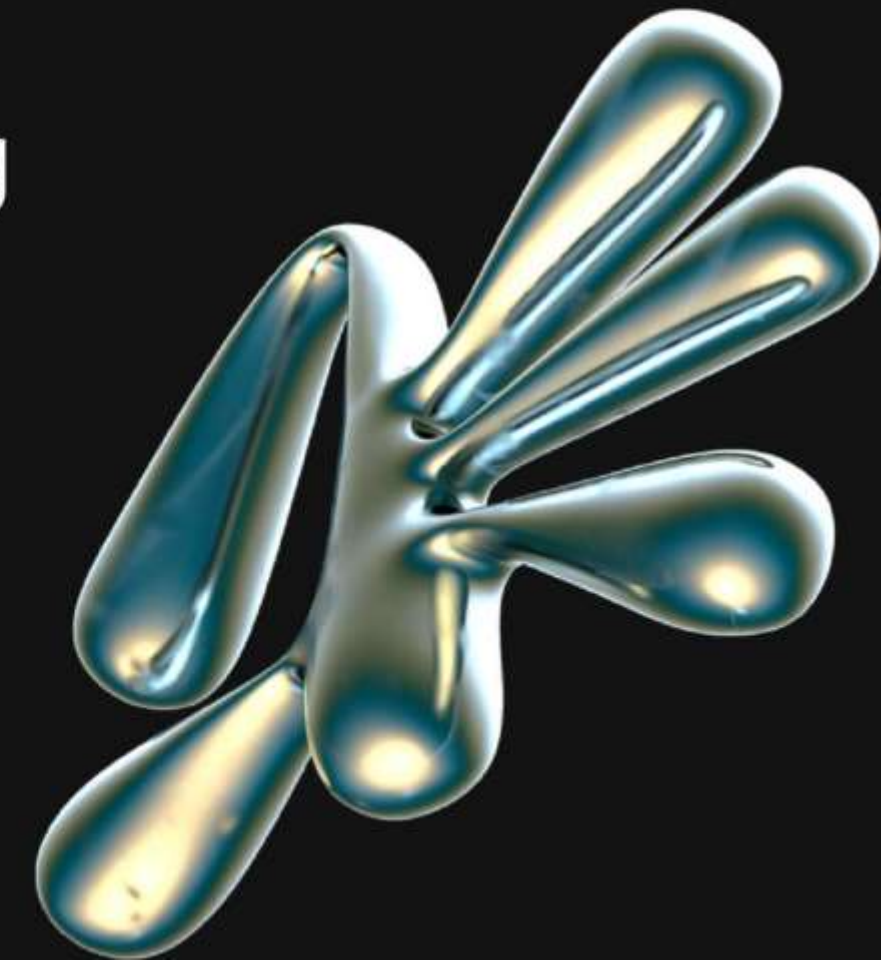
FREDDY MERNANDA

MULKI DJENFIK

STACK

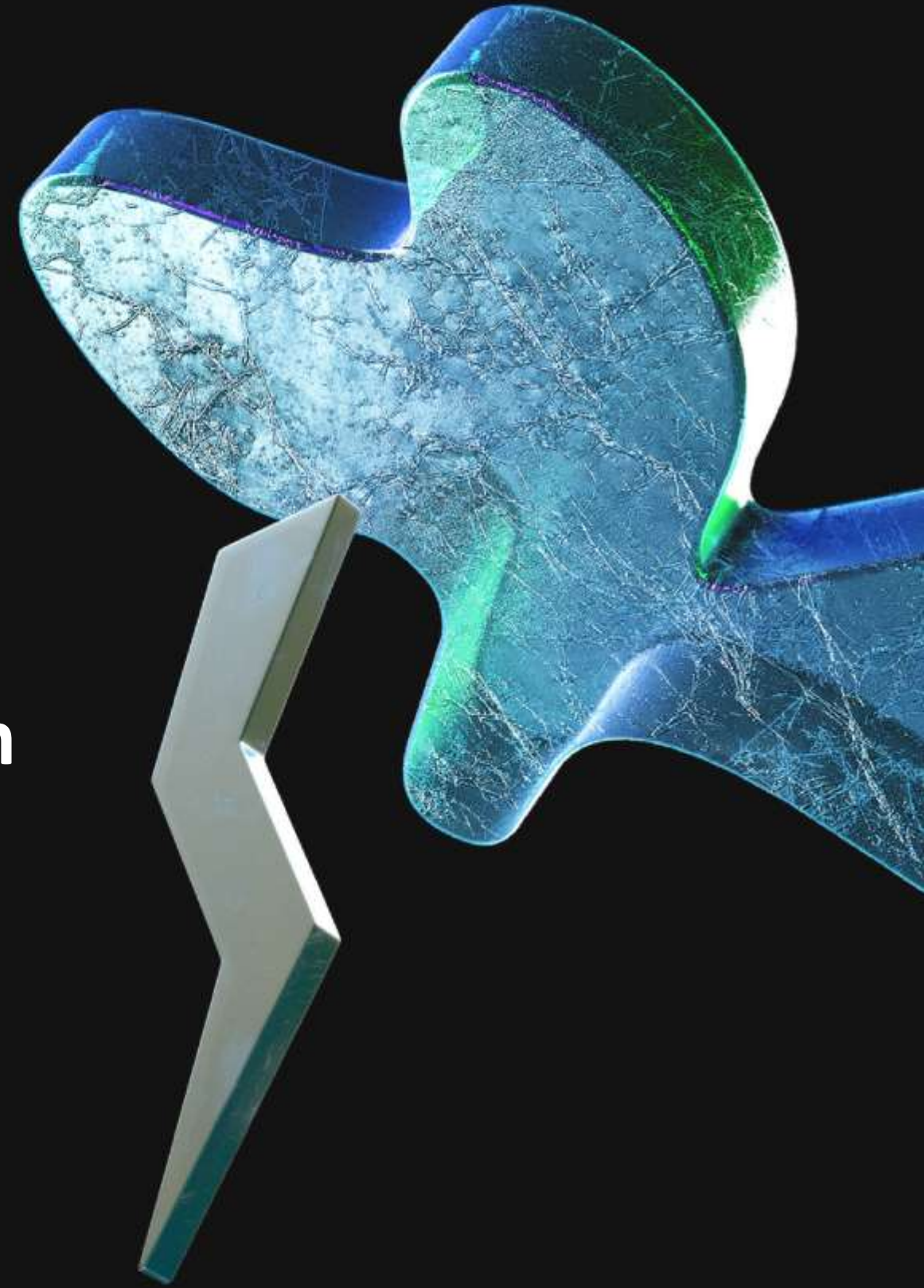
stack adalah salah satu struktur data yang digunakan untuk menyimpan sekumpulan objek ataupun variabel. Sesuai namanya yaitu stack, tidak heran apabila objek yang terkumpul terlihat seperti tumpukan.

Nah, karakteristik stack sendiri bersifat LIFO (last in first out). Artinya, data yang terakhir masuk merupakan data yang akan keluar terlebih dahulu. Seperti halnya tumpukan pada umumnya, misalnya tumpukan buku, yang di atas atau yang terakhir masuk harus dikeluarkan terlebih dahulu untuk mendapatkan buku yang berada di tumpukan bawah.



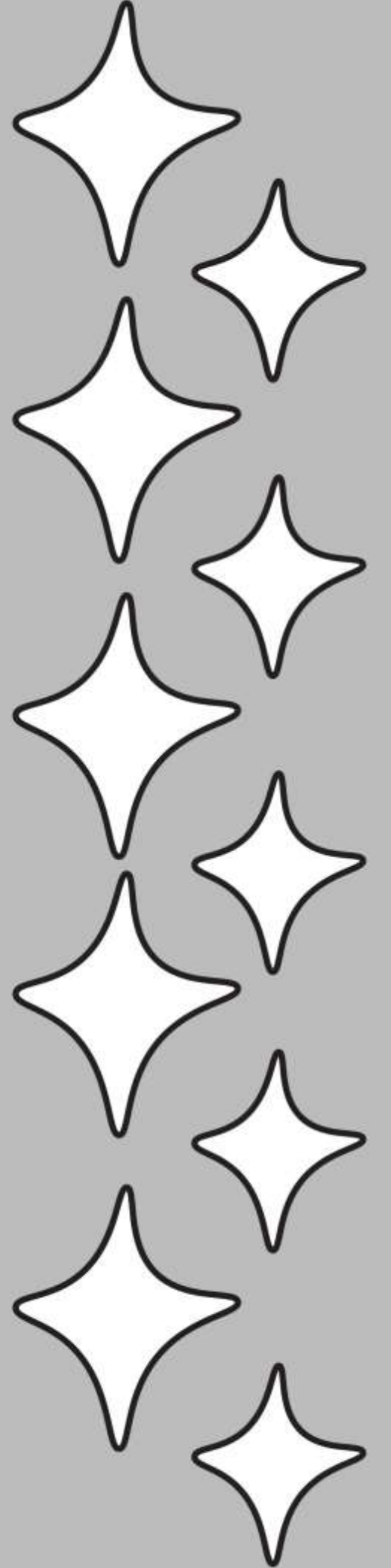
FUNGSI STACK

- Digunakan untuk menyimpan sementara data atau informasi dalam suatu program.
- Digunakan dalam algoritma pemrosesan data seperti algoritma infix-to-postfix, evaluasi ekspresi postfix, dan algoritma pembalikan string.
- Digunakan dalam implementasi algoritma depth-first search (DFS) pada struktur data graf atau pohon.



JENIS PEMROGRAMAN STACK

1. **Push:** operasi ini digunakan untuk menambahkan data baru ke dalam stack.
2. **Pop:** operasi ini digunakan untuk menghapus elemen teratas dari stack.
3. **Top:** operasi ini digunakan untuk mengakses elemen teratas dari stack tanpa menghapusnya.
4. **Size:** operasi ini digunakan untuk mendapatkan jumlah elemen dalam stack.
5. **Empty:** operasi ini digunakan untuk memeriksa apakah stack kosong atau tidak.
6. **Peek:** operasi ini mirip dengan operasi top, namun menggunakan istilah yang berbeda. Operasi peek digunakan untuk mengakses elemen teratas dari stack tanpa menghapusnya.
7. **Search:** operasi ini digunakan untuk mencari elemen tertentu dalam stack dan mengembalikan jaraknya dari elemen teratas (dalam urutan nomor, bukan dalam ukuran byte). Jika elemen tidak ditemukan, maka akan mengembalikan nilai -1.



CONTOH PEMROGRAMAN STACK



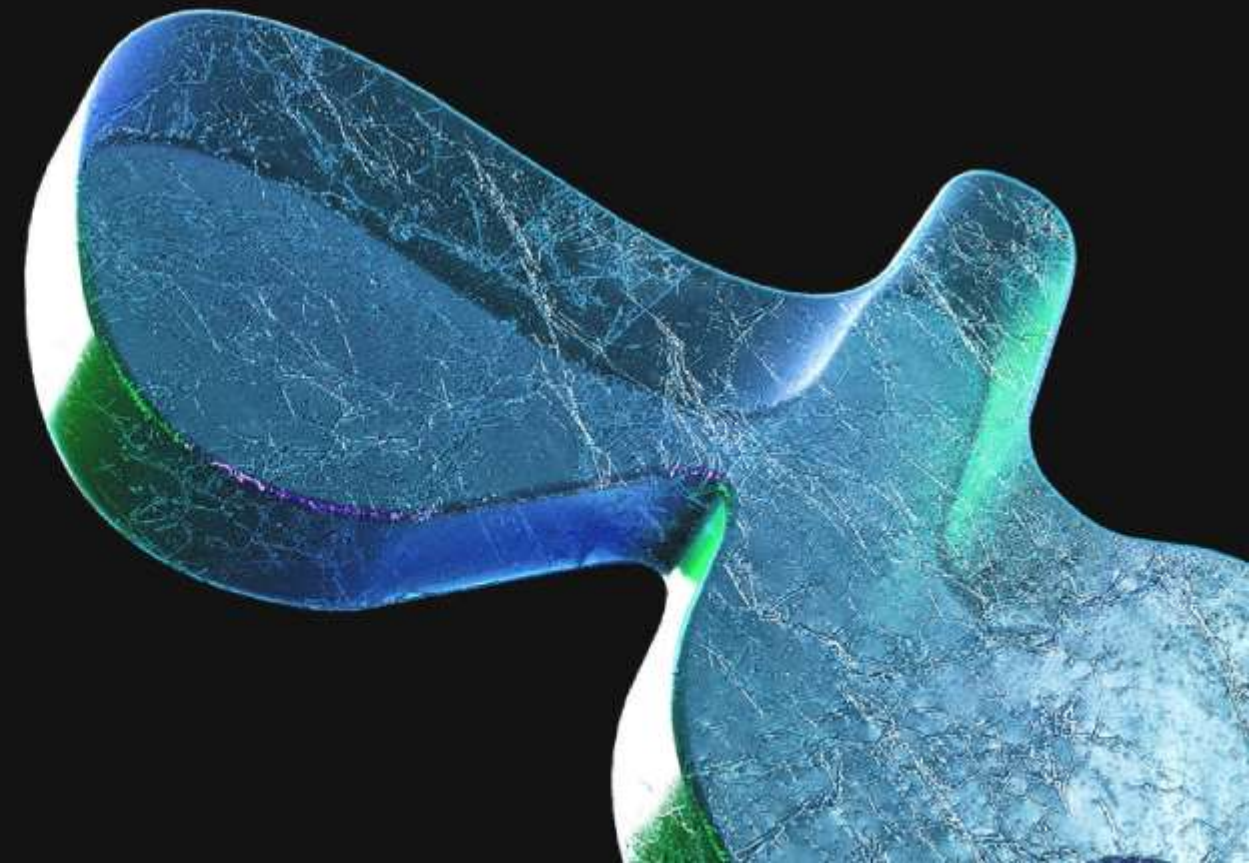
```
Jimas\OneDrive\Dokumen\Stack.cpp - [Executing] - Dev-C++ 5.11
Search View Project Execute Tools AStyle Window Help
(globals)
Stack.cpp
1 #include <iostream>
2 #include <stack>
3
4 int main() {
5     std::stack<int> stack;
6
7     // Menambahkan elemen ke dalam stack
8     stack.push(1);
9     stack.push(2);
10    stack.push(3);
11
12    // Mengambil elemen paling atas (tanpa menghapusnya)
13    std::cout << stack.top() << std::endl; // Output: 3
14
15    // Menghapus elemen paling atas
16    stack.pop();
17
18    // Mengambil elemen paling atas setelah dihapus
19    std::cout << stack.top() << std::endl; // Output: 2
20
21    // Mengecek apakah stack kosong atau tidak
22    std::cout << stack.empty() << std::endl; // Output: 0 (false)
23
24    return 0;
25 }
26
```

Pada contoh di samping, pertama-tama kita menginclude header file stack untuk menggunakan class stack yang telah disediakan oleh library STL (Standard Template Library) pada C++. Kemudian kita membuat objek stack dengan tipe data int menggunakan konstruktor default `std::stack<int> stack;`

Selanjutnya, kita menambahkan elemen ke dalam stack menggunakan method `push()` dengan parameter nilai yang ingin ditambahkan. Kemudian kita mengambil nilai elemen paling atas pada stack tanpa menghapusnya menggunakan method `top()`. Setelah itu, kita menghapus elemen paling atas menggunakan method `pop()`. Kemudian kita mengambil elemen paling atas lagi setelah elemen sebelumnya dihapus. Terakhir, kita memeriksa apakah stack kosong atau tidak menggunakan method `empty()`. Method `empty()` akan mengembalikan nilai `true` jika stack kosong, dan `false` jika tidak kosong.

DOUBLE STACK

Double Stack adalah struktur data yang mirip dengan **stack**, namun dengan dua ujung. Dua ujung ini memungkinkan penambahan dan penghapusan elemen pada kedua ujung, sehingga kita dapat memasukkan elemen pada ujung pertama atau kedua dan menghapus elemen pada ujung pertama atau kedua juga.



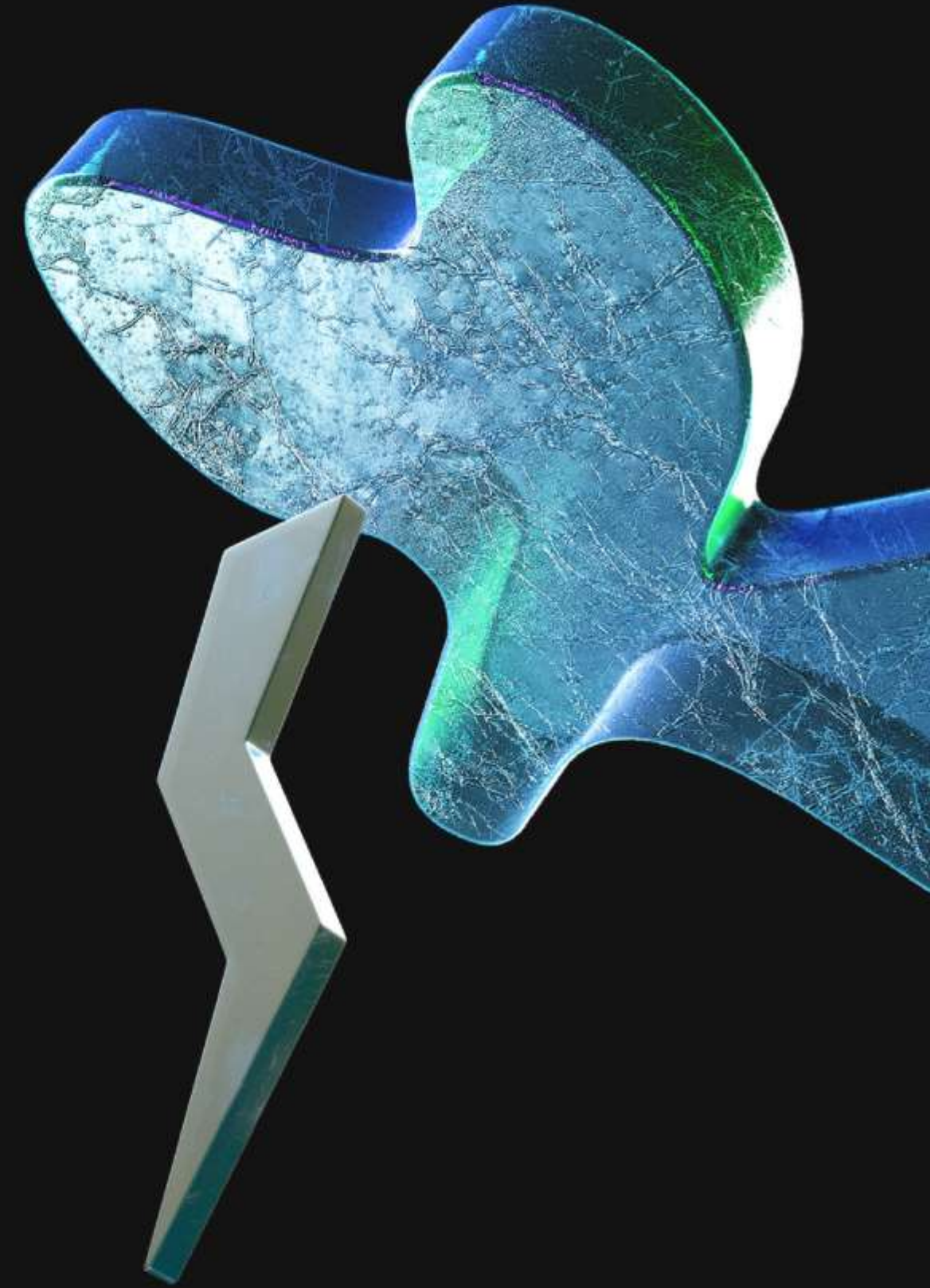
FUNGSI DOUBLE STACK

- Digunakan dalam pengolahan data dengan dua jenis atau kategori yang berbeda.
- Digunakan dalam implementasi algoritma merge sort untuk menggabungkan dua set data yang telah diurutkan.
- Digunakan dalam pengimplementasian algoritma pemrosesan data seperti algoritma postfix evaluation, infix-to-postfix conversion, dan sebagainya.



JENIS OPERASI DOUBLE STACK

- **Push left:** operasi ini digunakan untuk menambahkan elemen ke dalam stack kiri dari sisi kiri (bottom).
- **Push right:** operasi ini digunakan untuk menambahkan elemen ke dalam stack kanan dari sisi kanan (bottom).
- **Pop left:** operasi ini digunakan untuk menghapus elemen teratas dari stack kiri.
- **Pop right:** operasi ini digunakan untuk menghapus elemen teratas dari stack kanan.
- **Peek left:** operasi ini digunakan untuk mengakses elemen teratas dari stack kiri tanpa menghapusnya.



- Peek right: operasi ini digunakan untuk mengakses elemen teratas dari stack kanan tanpa menghapusnya.
- Is empty: operasi ini digunakan untuk memeriksa apakah double stack kosong atau tidak.
- Is full: operasi ini digunakan untuk memeriksa apakah double stack penuh atau tidak.
- Clear: operasi ini digunakan untuk menghapus semua elemen dari kedua stack.
- Size: operasi ini digunakan untuk mendapatkan jumlah elemen dalam kedua stack.



CONTOH DOUBLE STACK

KELEBIHAN

- Membantu mengelola data dengan metode LIFO
- Secara otomatis membersihkan objek
- Tidak mudah rusak
- Ukuran variabel tidak dapat diubah
- Mengontrol memori secara mandiri



Kekurangan

- Memori stack sangat terbatas.
- Ada kemungkinan stack akan meluap atau overflow jika terlalu banyak objek.
- Tidak memungkinkan akses acak, karena harus mengeluarkan tumpukan paling atas terlebih dahulu untuk mengakses tumpukan paling bawah.

