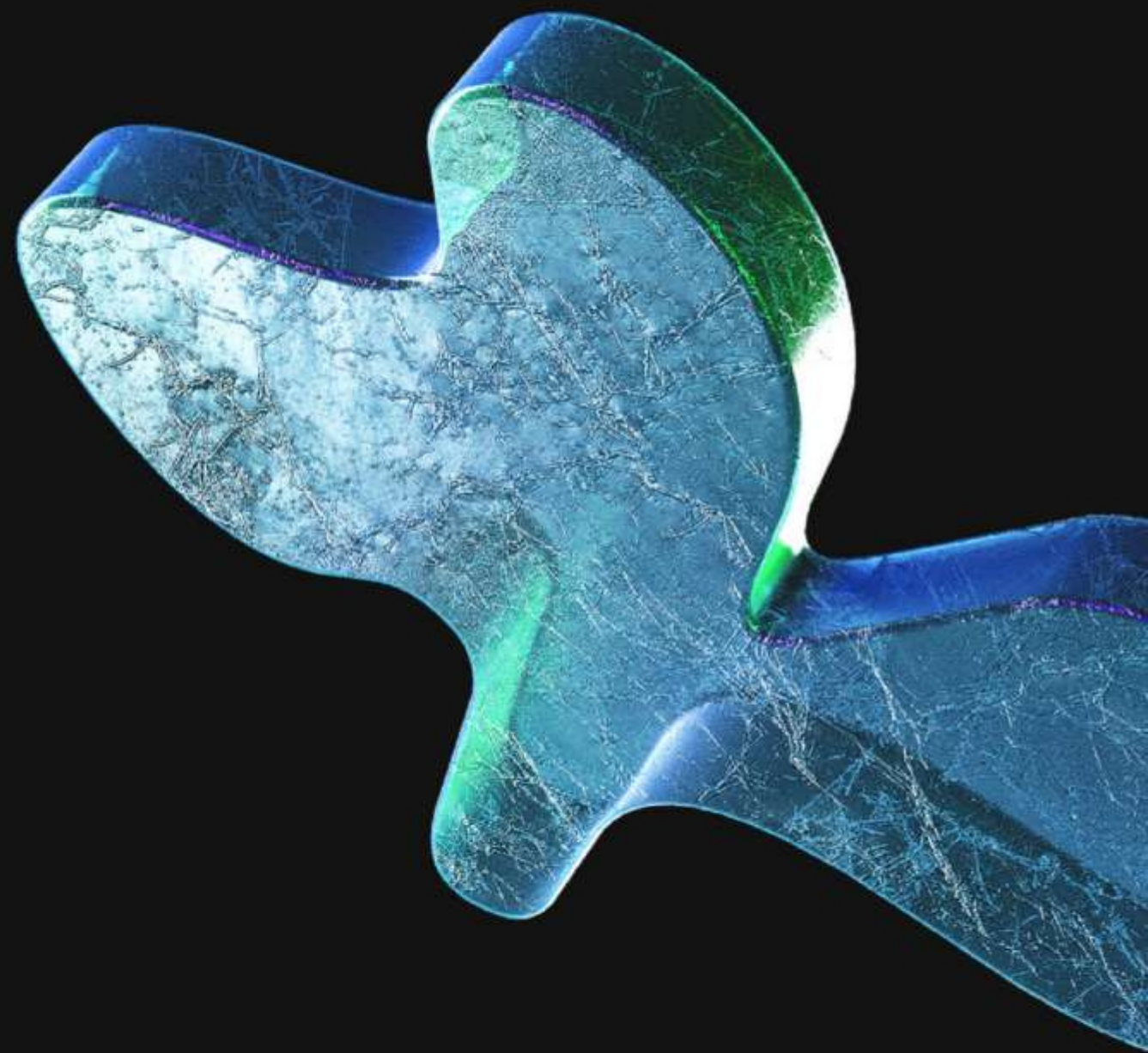




STACK DAN DOUBLE STACK



KELOMPOK 4



MUHAMMAD RAYYAN GHIFARI

SADIN YUSUF ARDIKA

DIMAS AUFA MUHAMMAD ZAKI

GALIH DWI NUGROHO

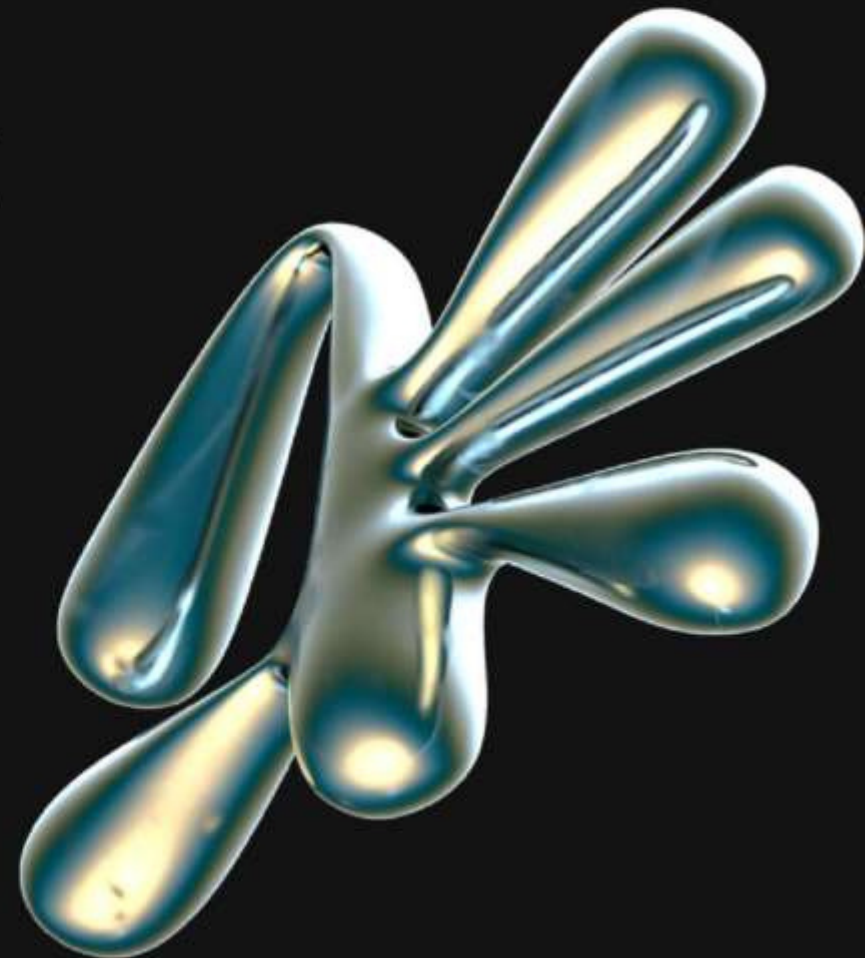
FREDDY MERNANDA

MULKI DJENFIK

STACK

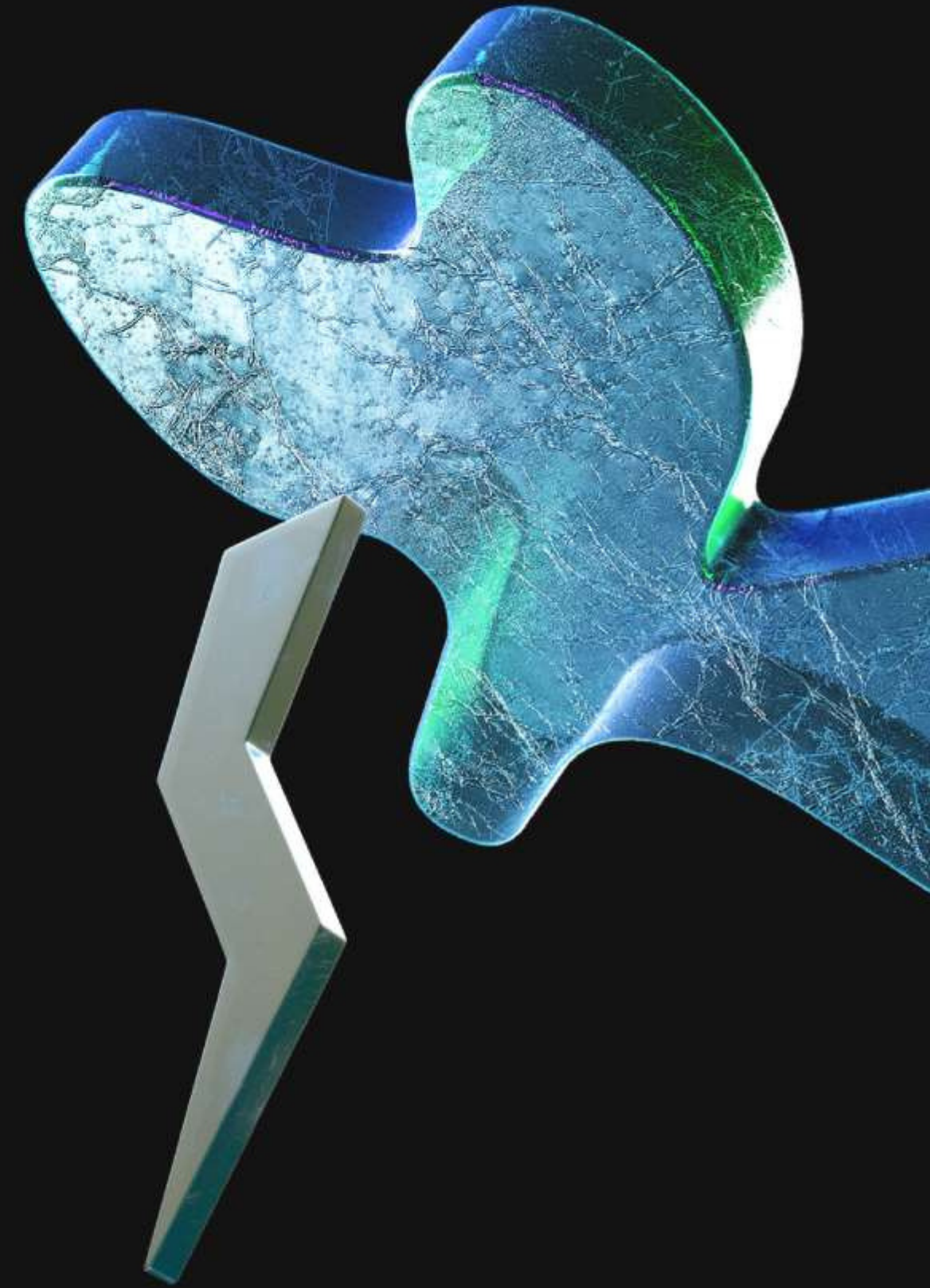
stack adalah salah satu struktur data yang digunakan untuk menyimpan sekumpulan objek ataupun variabel. Sesuai namanya yaitu stack, tidak heran apabila objek yang terkumpul terlihat seperti tumpukan.

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.



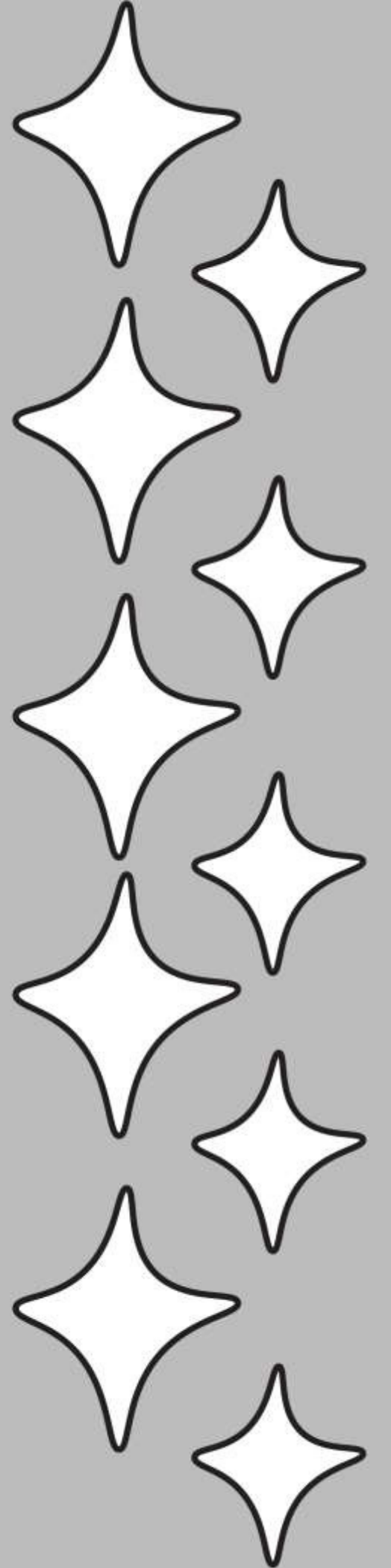
FUNGSI STACK

- Digunakan untuk menyimpan sementara data atau informasi dalam suatu program.
- Digunakan dalam Reversing data
- Dapat digunakan sebagai manajemen memori pada program komputer
- Digunakan dalam pemrosesan tumpukan data



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

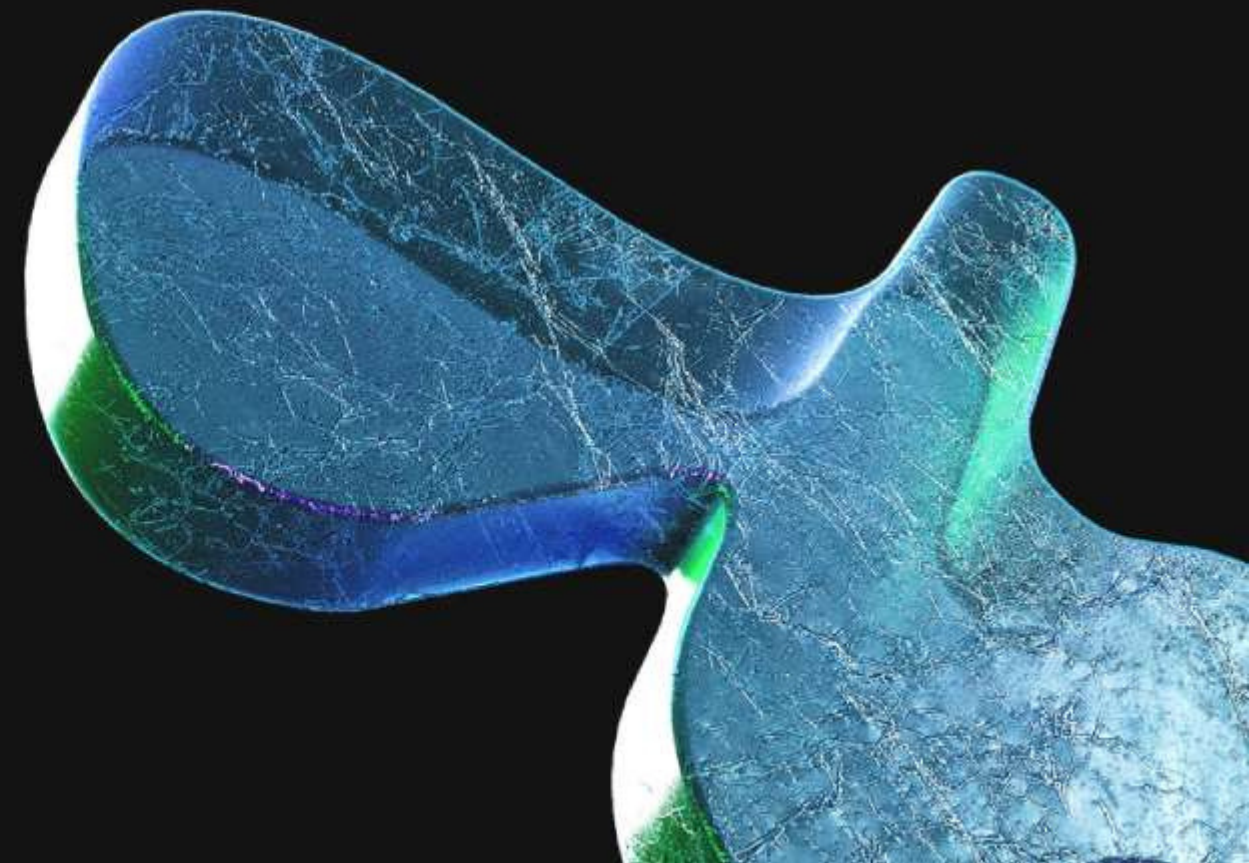
```
1  #include <iostream>
2  #include <stdlib.h>
3  #include <conio.h>
4  #define MAX 10 // batas stack yang ingin dibuat, boleh lebih
5  using namespace std;
6
7  int top=-1, Stack[MAX];
8
9  void push(){
10     if(top == MAX-1){
11         cout << ">> Stack sudah Penuh !" << endl;
12     }else{
13         top++;
14         cout << ">> Masukan Data : ";
15         cin >> Stack[top];
16
17         cout << "Data [" << Stack[top] << "] Telah Ditambah !" << endl;
18     }
19 }
20
21 void pop(){
22     if(top == -1){
23         cout << ">> Stack Kosong !" << endl;
24     }else{
25         cout << "nData [" << Stack[top] << "] pada index ke '" << top << "' dalam Stack Diambil !" << endl;
26         Stack[top--];
27     }
28 }
29
30 int clearr(){
31     return top = -1;
32 }
33
34 void Print(){
35     if (top == -1) {
36         cout << "
37         cout << "n===== " << endl;
```

```
38         cout << "
39     }
40     else {
41         cout << "
42         cout << "n===== " << endl;
43     for (int i = top; i >= 0; i--){
44         cout << "
45     }
46     cout << "===== " << endl;
47 }
48
49
50 int main()
51 {
52     int choose;
53     do {
54         Print();
55         cout << "n1. Push"
56         << "n2. Pop"
57         << "n3. Clear"
58         << "n4. Exit"
59         << "nPergerakan ( TOP ) : " << top
60         << "nn>> Choose : "; cin >> choose;
61     switch (choose){
62         case 1:
63             push();getch();
64             break;
65         case 2:
66             pop();getch();
67             break;
68         case 3:
69             clearr();
70             cout << "nClear Succes" << endl;
71             getch();
72             break;
73
74         default:
```

```
75         cout << "Pilihan tidak ada\nMasukan pilihan sesuai dengan daftar" << endl;
76         getch();
77         break;
78     }
79     system("cls");
80 }while(choose !=4);
81 }
```


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.



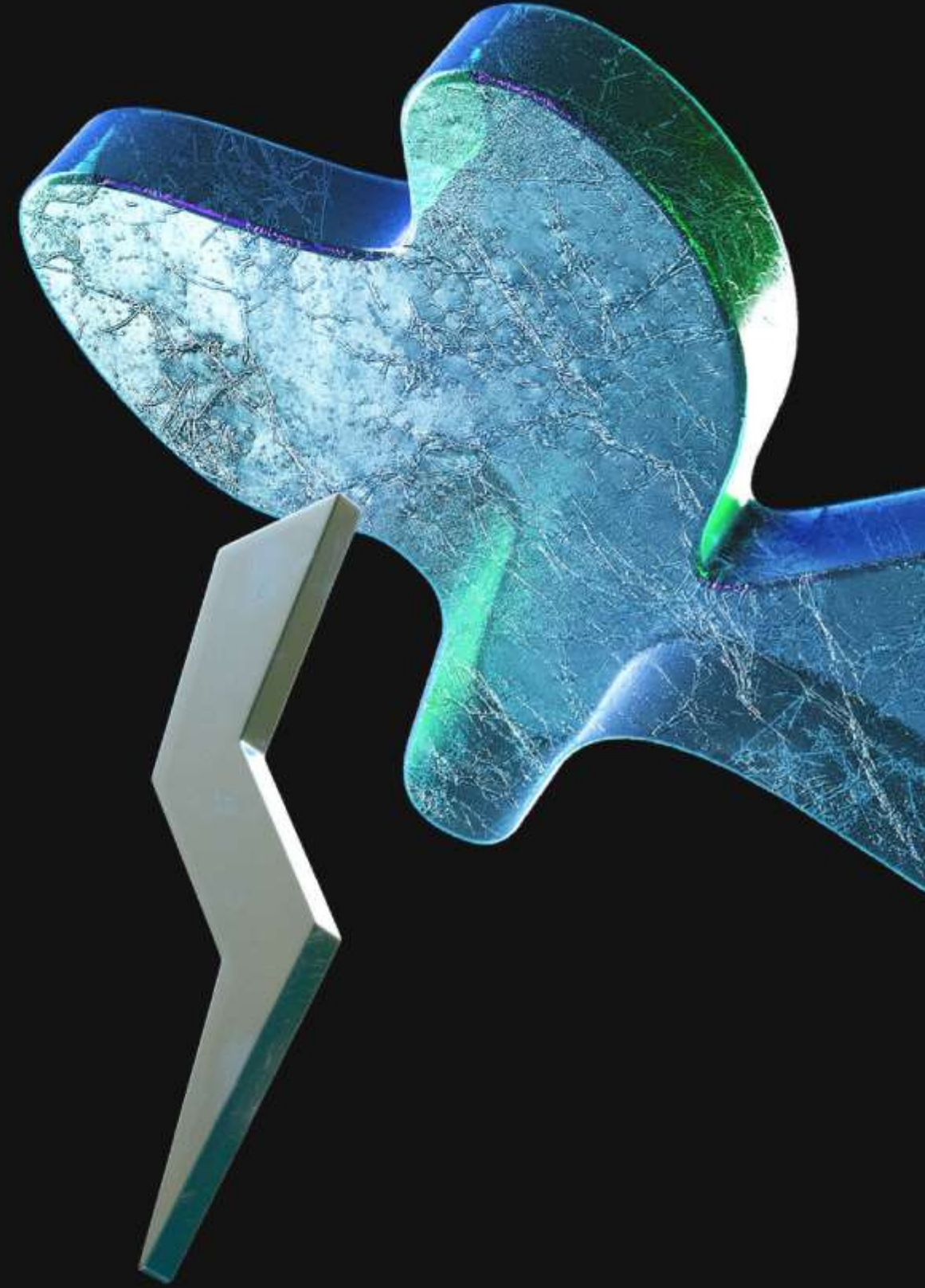
FUNGSI DOUBLE STACK

- 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 mengevaluasi ekspresi matematika.



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.



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.

