

Pengenalan pada stack (tumpukan)

- Apa itu stack?
Tumpukan adalah ilustrasi dari struktur data linier Last In First Out (LIFO). Hal ini menunjukkan bahwa penambahan terbaru ke tumpukan adalah yang pertama kali diambil. Tumpukan digunakan secara luas untuk mengimplementasikan rekursi, operasi undo/redo, dan backtracking.
- Struktur data pada stack (tumpukan)
Tumpukan dapat direpresentasikan sebagai daftar elemen, di mana setiap elemen memiliki penunjuk ke elemen berikutnya dalam daftar. Bagian atas tumpukan adalah elemen pertama dalam daftar, dan bagian bawah tumpukan adalah elemen terakhir dalam daftar.
- Operasi pada stack (tumpukan)
Empat operasi dasar dapat dilakukan pada sebuah tumpukan:
 - * Push: Menambahkan komponen ke bagian atas tumpukan.
 - * Pop: Metode ini menghilangkan elemen dari bagian atas tumpukan.
 - * Peek: Tanpa menghapus elemen teratas dari tumpukan, fungsi ini mengembalikannya.
 - * IsEmpty: Memverifikasi bahwa tumpukan kosong.
- Aplikasi pada stack (tumpukan)
Tumpukan digunakan dalam berbagai aplikasi dalam ilmu komputer dan rekayasa perangkat lunak. Berikut ini adalah beberapa contoh dari penggunaan tumpukan:
 1. Evaluasi ekspresi aritmatika¹²³
 2. Penelusuran mundur, yaitu untuk memeriksa kecocokan tanda kurung dalam sebuah ekspresi²⁴
 3. Konversi dari satu bentuk ekspresi ke bentuk lainnya²
 4. Manajemen memori yang sistematis²
 5. Penguraian dan analisis sintaksis bahasa pemrograman⁴
 6. Mengaktifkan fungsionalitas undo/redo dalam program⁴³
 7. Menyimpan pemanggilan fungsi dan statusnya, yang membantu dalam implementasi pemanggilan fungsi rekursif yang efisien⁴
 8. Tumpukan panggilan dalam bahasa pemrograman³
 9. Evaluasi ekspresi dalam matematika³
 10. Alokasi memori oleh sistem operasi saat menjalankan sebuah proses²
 11. Menyimpan log panggilan, email, dan foto Google di galeri mana pun⁴
 12. Unduhan dan pemberitahuan YouTube juga ditampilkan dalam format LIFO (yang terbaru muncul lebih dulu)⁴
 13. Tumpukan pemasaran, yang merupakan kumpulan alat dan teknologi yang digunakan oleh pemasar untuk mengeksekusi dan menganalisis aktivitas pemasaran⁵.

¹ <https://www.javatpoint.com/applications-of-stack-in-data-structure>

² <https://byjus.com/gate/stack-and-its-applications/>

³ <https://techalmirah.com/5-real-world-applications-of-the-stack-data-structure/>

⁴ <https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-stack/>

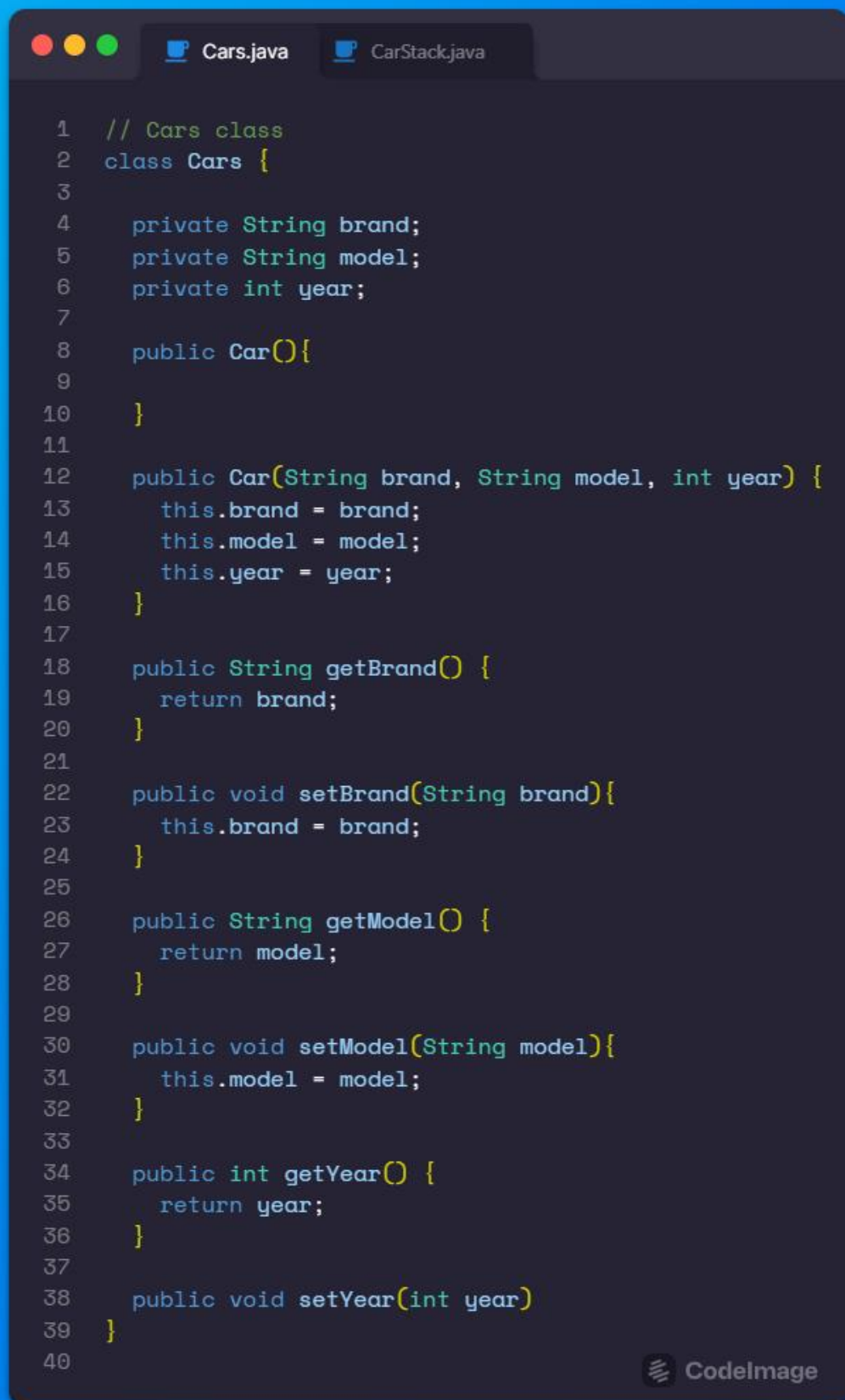
⁵ <https://www.techtarget.com/searchapparchitecture/definition/software-stack>

Secara umum, stack digunakan di semua aplikasi di mana data harus disimpan dan diambil di urutan terakhir¹.


Pengimplementasian pada stack (tumpukan)

- Kelas dari stack (tumpukan)
Bahasa pemrograman Java menyediakan kelas `Stack`, yang dapat digunakan untuk mengimplementasikan tumpukan.
- Fungsi atau Metode pada stack (tumpukan)
Kelas `Stack` memiliki metode untuk melakukan empat operasi dasar tumpukan: `push()`, `pop()`, `peek()`, dan `isEmpty()`.
- Contoh dari pengimplementasian dari stack (tumpukan)

¹ <https://www.javatpoint.com/applications-of-stack-in-data-structure>



```
1 // Cars class
2 class Cars {
3
4     private String brand;
5     private String model;
6     private int year;
7
8     public Car(){
9
10    }
11
12    public Car(String brand, String model, int year) {
13        this.brand = brand;
14        this.model = model;
15        this.year = year;
16    }
17
18    public String getBrand() {
19        return brand;
20    }
21
22    public void setBrand(String brand){
23        this.brand = brand;
24    }
25
26    public String getModel() {
27        return model;
28    }
29
30    public void setModel(String model){
31        this.model = model;
32    }
33
34    public int getYear() {
35        return year;
36    }
37
38    public void setYear(int year)
39 }
40
```

 CodeImage

```
1  import java.util.Stack;
2
3  public class CarStack {
4
5      public static void main(String[] args) {
6          // Create a stack to store cars
7          Stack<Car> carStack = new Stack<>();
8
9          // Add some cars to the stack
10         carStack.push(new Car("Toyota", "Corolla", 2023));
11         carStack.push(new Car("Honda", "Civic", 2022));
12         carStack.push(new Car("Tesla", "Model 3", 2021));
13
14         // Print the cars in the stack
15         while (!carStack.isEmpty()) {
16             Car car = carStack.pop();
17             System.out.println("Berikut adalah Brand dari
mobil " + car.getBrand() + " dengan model " +
car.getModel() + " dibuat pada tahun " + car.getYear());
18         }
19     }
20 }
```

Latihan

- Implementasikan stack dalam bentuk linked list (senarai berantai)
- Implementasikan stack dalam bentuk array list (daftar larik)
- Buatlah sebuah program untuk melakukan reverse string menggunakan stack