

# **Praktikum Algoritma Struktur Data**

## **Praktikum 04: Overview Stack**



**Oleh:**

Bintang Bimantara/ 5223600025

**Program Studi Sarjana Terapan Teknologi Game**

**Departemen Teknologi Multimedia Kreatif**

**Politeknik Elektronika Negeri Surabaya**

**2024**

## 1. Implementasikan Stack Menggunakan Linked List

Kelas `ListStack` merupakan implementasi Stack, dengan menggunakan operasi dasar seperti `push` (menambahkan elemen ke Stack), `pop` (menghapus elemen dari Stack), `peek` (melihat elemen teratas tanpa menghapusnya), `size` (mendapatkan ukuran Stack), dan `empty` (memeriksa apakah Stack kosong).

Output

Clear

```
./tmp/LWIadHzS6G.o
Top element: 2
Size of stack: 2
Top element after pop: 1

=== Code Execution Successful ===
```

## 2. Memeriksa Keseimbangan Tanda Kurung

Memeriksa keseimbangan tanda kurung dalam sebuah ekspresi menggunakan Stack dengan mendorong tanda kurung pembuka ke dalam Stack dan mencocokkannya dengan tanda kurung penutup

```
Output Clear
/tmp/hfJZDvScYn.o
Expression: {} is balanced
Expression: {} is not balanced

=== Code Execution Successful ===
```

### 3. Konversi Infix ke Postfix

Mengonversi ekspresi matematis dari notasi infix ke postfix menggunakan Stack untuk menangani prioritas operator sesuai dengan aturan precedence.

```
Output Clear
/tmp/TegVwePkkp.o
Infix expression: 10+(3)*5/(16-4)
Postfix expression: 1035*164-/+

=== Code Execution Successful ===
```

### 4. Konversi Infix ke Prefix

Mengonversi ekspresi matematis dari notasi infix ke prefix dengan membalik ekspresi infix, mengonversinya ke postfix, lalu membalik kembali hasilnya

Output

Clear

```
▲ /tmp/4oaLN7Wx06.o
Infix expression: a+b*(c^d-e)^(f+g*h)-i+
Prefix expression: -(^)+a*bcd+e^f*gh-+i

=== Code Execution Successful ===
```

## 5. Evaluasi Ekspresi Postfix

Mengevaluasi ekspresi postfix dengan mendorong operand ke dalam Stack dan melakukan operasi aritmatika sesuai operator yang ditemukan

Output

```
/tmp/9AjMezmHrT.o
Postfix expression: 6523+8*+3+*
Result: 288

=== Code Execution Successful ===
```

## 6. Palindrome String menggunakan Stack

Untuk memeriksa apakah sebuah string merupakan palindrome atau tidak dengan membandingkan setengah karakter pertama yang didorong ke dalam Stack dengan setengah karakter kedua

## Output

Clear

▲ /tmp/8P0bdTwdj0.o

Masukkan sebuah string: ak u k a

"ak u k a" is a palindrome

=== Code Execution Successful ===