

**LAPORAN PRAKTIKUM  
BASIS DATA**

**PRAKTIKUM 7  
Stack**



**Oleh:  
JESSICA AMELIA  
2341760185  
SIB 1A/16**

**PROGRAM STUDI SISTEM INFORMASI BISNIS  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2023/2024**

## Jobsheet 7

### Praktikum 1

1. Pada method pop(), mengapa diperlukan pemanggilan method isEmpty()?

**Jawab :** Karena pada method pop(), pemanggilan method isEmpty() diperlukan untuk memeriksa apakah stack kosong sebelum mencoba untuk menghapus elemen dari stack. Jika stack kosong dan kita mencoba untuk menghapus elemen dari stack yang kosong, itu akan menyebabkan kesalahan atau kesalahan dalam program. Oleh karena itu, dengan memanggil isEmpty() sebelum pop(), kita dapat memastikan bahwa hanya akan mencoba untuk menghapus elemen dari stack jika stack tidak kosong.

2. Pada method push(), mengapa diperlukan pemanggilan method isFull()?

**Jawab :** Pada method push(), pemanggilan method isFull() diperlukan untuk memeriksa apakah stack sudah penuh sebelum mencoba untuk menambahkan elemen baru ke dalamnya. Jika stack sudah penuh dan kita mencoba untuk menambahkan elemen baru, itu akan menyebabkan overflow atau error dalam program. Dengan memanggil isFull() sebelum push(), kita dapat memastikan bahwa hanya akan mencoba untuk menambahkan elemen baru ke dalam stack jika masih ada ruang kosong.

3. Jelaskan perbedaan antara method peek() dengan method pop() pada class Stack.

**Jawab :**

- Method peek(): Method ini digunakan untuk melihat elemen teratas (atau elemen terakhir) dari stack tanpa menghapusnya dari stack. Dengan kata lain, method peek() hanya mengembalikan nilai dari elemen teratas tanpa mengubah struktur stack.
- Method pop(): Method ini digunakan untuk menghapus dan mengembalikan elemen teratas (atau elemen terakhir) dari stack. Setelah pemanggilan pop(), elemen yang dihapus tidak lagi ada di stack.

### Praktikum 2

1. Apa fungsi kode program berikut

```
c = Q.charAt(i);
```

**Jawab :**

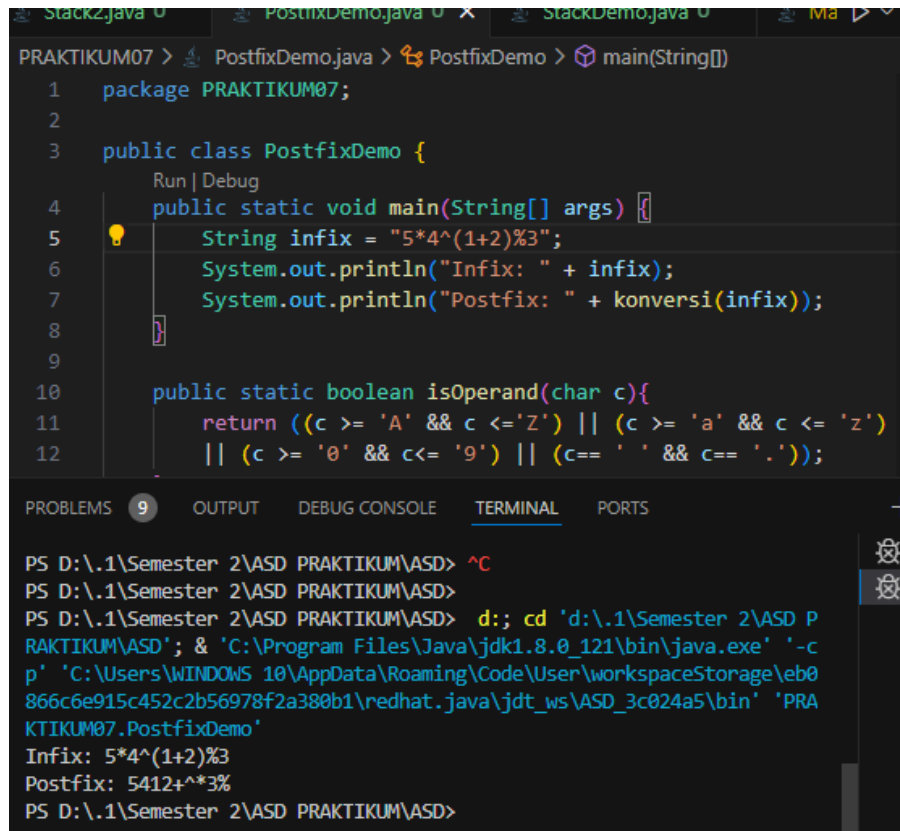
Kode program `c = Q.charAt(i);` berfungsi untuk mengambil karakter pada posisi tertentu dalam string Q dan menyimpannya ke dalam variabel c.

c: Ini adalah variabel yang akan menyimpan karakter yang diambil dari string Q.

Q adalah variabel bertipe String yang berisi text dari mana karakter akan diambil.

charAt(i) adalah method (fungsi) yang digunakan untuk mengambil karakter pada posisi tertentu (i) dari String Q. Penomoran dimulai dari 0, sehingga `Q.charAt(0)` akan mengembalikan karakter pertama di dalam String Q.

2. Jalankan kembali program tersebut, masukkan ekspresi  $5 \cdot 4^{(1+2)} \% 3$ . Tampilkan hasilnya!



```
PRAKTIKUM07 > PostfixDemo.java > PostfixDemo > main(String[])
1 package PRAKTIKUM07;
2
3 public class PostfixDemo {
4     public static void main(String[] args) {
5         String infix = "5*4^(1+2)%3";
6         System.out.println("Infix: " + infix);
7         System.out.println("Postfix: " + konversi(infix));
8     }
9
10    public static boolean isOperand(char c){
11        return ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'))
12            || (c >= '0' && c <= '9') || (c == '.' && c == '.');
13    }
14 }
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD> ^C
PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD>
PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD> d;; cd 'd:\.1\Semester 2\ASD P
RAKTIKUM\ASD'; & 'C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' '-c
p' 'C:\Users\WINDOWS 10\AppData\Roaming\Code\User\workspaceStorage\eb0
866c6e915c452c2b56978f2a380b1\redhat.java\jdt_ws\ASD_3c024a5\bin' 'PRA
KTIKUM07.PostfixDemo'
Infix: 5*4^(1+2)%3
Postfix: 5412+^*3%
PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD>
```

## Tugas

implementasikan kode program untuk menghitung hasil dari persamaan yang sudah dalam notasi infix dengan memanfaatkan stack

```

public static int hitungPostfix(String postfix) {
    Stack2 operands = new Stack2(postfix.length());

    for (int i = 0; i < postfix.length(); i++) {
        char c = postfix.charAt(i);
        if (isOperand(c)) {
            operands.push((char) (c - '0'));
        } else if (isOperator(c)) {
            int operand1 = operands.pop();
            int operand2 = operands.pop();
            int result = 0;
            switch (c) {
                case '+':
                    result = operand2 + operand1;
                    break;
                case '-':
                    result = operand2 - operand1;
                    break;
                case '*':
                    result = operand2 * operand1;
                    break;
                case '/':
                    result = operand2 / operand1;
                    break;
                case '%':
                    result = operand2 % operand1;
                    break;
                case '^':
                    result = (int) Math.pow(operand2, operand1);
                    break;
            }
            operands.push((char) result);
        }
    }
    return operands.pop();
}

```

in' 'PRAKTIKUM07.PostfixDemo'

Infix: 5\*4^(1+2)%3

Postfix: 5412+^\*3%

Hasil: 2

PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD>

main\* 0 9 0 Java: Ready

Spaces: 4 UTF