

LAPORAN PRAKTIKUM
MATA KULIAH PRAKTIKUM ALGORITMA STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN 9 : STACK



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

PRAKTIKUM 1

Kode program class Stack29 :

```
1 public class Stack29 {
2     int data[];
3     int size;
4     int top;
5
6     public Stack29(int size) {
7         this.size = size;
8         data = new int[size];
9         top = -1;
10    }
11
12    public boolean IsFull() {
13        if (top == size-1) {
14            return true;
15        } else {
16            return false;
17        }
18    }
19
20    public boolean IsEmpty() {
21        if (top == -1) {
22            return true;
23        } else {
24            return false;
25        }
26    }
27
28    public void push(int dt) {
29        if (!IsFull()) {
30            top++;
31            data[top] = dt;
32        } else {
33            System.out.println("Stack penuh");
34        }
35    }
36
37    public void pop() {
38        if (!IsEmpty()) {
39            int x = data[top];
40            top--;
41            System.out.println("Data yang dikeluarkan dari stack : " + x);
42        } else {
43            System.out.println("Stack masih kosong");
44        }
45    }
46
47    public void peek() {
48        System.out.println("Elemen teratas stack : " + data[top]);
49    }
50
51    public void print() {
52        System.out.println("Isi stack : ");
53        for (int i = top; i >= 0; i--) {
54            System.out.println(data[i] + " ");
55        }
56        System.out.println("");
57    }
58
59    public void clear() {
60        if (!IsEmpty()) {
61            for (int i = top; i >= 0; i--) {
62                top--;
63            }
64            System.out.println("Stack sudah dikosongkan");
65        } else {
66            System.out.println("Stack masih kosong");
67        }
68    }
69 }
70 }
```

Kode program class DemoStack29 :

```
1 public class StackDemo29 {
2     public static void main(String[] args) {
3         Stack29 stack29 = new Stack29(10);
4         stack29.push(8);
5         stack29.push(12);
6         stack29.push(18);
7         stack29.print();
8         stack29.pop();
9         stack29.peek();
10        stack29.pop();
11        stack29.push(-5);
12        stack29.print();
13    }
14 }
15 }
```

Output :

```
Isi stack :
18
12
8

Data yang dikeluarkan dari stack : 18
Elemen teratas stack : 12
Data yang dikeluarkan dari stack : 12
Isi stack :
-5
8

PS D:\KULIAH\college\smt 2\29_yonanda_asd\P9>
```

Pertanyaan :

1. Pada method pop(), mengapa diperlukan pemanggilan method isEmpty()? Apa yang terjadi jika tidak ada pemanggilan isEmpty()?

Jawab : Operasi pop digunakan untuk mengambil elemen stack pada tumpukan paling atas (top stack). Pada method pop(), diperlukan pemanggilan method isEmpty() yang berfungsi untuk memeriksa apakah data stack masih kosong, dengan cara memeriksa top of stack, jika masih -1 maka berarti data stack masih kosong. Jika tidak ada method isEmpty(), maka dalam operasi tumpukan stack yang kosong, operasi pop akan tetap dilakukan, namun hal tersebut dapat menyebabkan kesalahan saat mencoba mengakses elemen yang tidak ada dalam stack kosong.

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

Jawab : Method peek() digunakan untuk memeriksa elemen paling atas, sedangkan method pop() berfungsi untuk mengambil elemen top stack pada stack. Jadi, peek() hanya dibaca saja (menampilkan data saja) tidak diikuti decrement top, yang berbeda dengan operasi pop().

PRAKTIKUM 2

Kode program class Pakaian29 :

```
1 public class Pakaian29 {
2     String jenis, warna, merk, ukuran;
3     double harga;
4
5     Pakaian29(String jenis, String warna, String merk, String ukuran, double harga) {
6         this.jenis = jenis;
7         this.warna = warna;
8         this.merk = merk;
9         this.ukuran = ukuran;
10        this.harga = harga;
11    }
12 }
13 }
```

Kode program class Stack29 :

```
1 public class Stack29 {
2     int size;
3     int top;
4     Object[] data;
5
6     public Stack29(int size) {
7         this.size = size;
8         data = new Object[size];
9         top = -1;
10    }
11
12    public boolean isEmpty() {
13        return top == -1;
14    }
15
16    public boolean isFull() {
17        return top == size - 1;
18    }
19
20    public void push(Object obj) {
21        if (isFull()) {
22            System.out.println("Stack is Full");
23            return;
24        }
25        data[++top] = obj;
26        System.out.println("Pushed: " + obj);
27    }
28
29    public void pop() {
30        if (isEmpty()) {
31            System.out.println("Stack is Empty");
32            return;
33        }
34        Object obj = data[top--];
35        System.out.println("Popped: " + obj);
36    }
37
38    public void peek() {
39        if (isEmpty()) {
40            System.out.println("Stack is Empty");
41            return;
42        }
43        System.out.println("Peek: " + data[top]);
44    }
45
46    public void print() {
47        System.out.println("Stack contents:");
48        for (int i = top; i >= 0; i--) {
49            System.out.println(data[i]);
50        }
51    }
52
53    public void clear() {
54        top = -1;
55        System.out.println("Stack cleared");
56    }
57 }
```

Kode program class StackMain29 :

```
1 import java.util.Scanner;
2
3 public class StackMain29 {
4     public static void main(String[] args) {
5         Scanner sc29 = new Scanner(System.in);
6         Stack29<P> stk = new Stack29(5);
7
8         char pilih;
9         do {
10             System.out.print("Jenis : ");
11             String jenis = sc29.nextLine();
12             System.out.print("Warna : ");
13             String warna = sc29.nextLine();
14             System.out.print("Merk : ");
15             String merk = sc29.nextLine();
16             System.out.print("Ukuran : ");
17             String ukuran = sc29.nextLine();
18             System.out.print("Harga : ");
19             double harga = sc29.nextDouble();
20
21             Pakaiac29 p = new Pakaiac29(jenis, warna, merk, ukuran, harga);
22             System.out.print("Apakah anda akan menambahkan data baru ke stack (y/n)? ");
23             pilih = sc29.next().charAt(0);
24             sc29.nextLine();
25             stk.push(p);
26         } while (pilih == 'y');
27
28         stk.print();
29         stk.pop();
30         stk.peek();
31         stk.print();
32     }
33 }
```

Output :

```
Jenis : Kaos
Warna : Hitam
Merk : Nevada
Ukuran : M
Harga : 85000
Apakah anda akan menambahkan data baru ke stack (y/n)? y
Jenis : Kemeja
Warna : Putih
Merk : Styves
Ukuran : XL
Harga : 127000
Apakah anda akan menambahkan data baru ke stack (y/n)? y
Jenis : Celana
Warna : Biru
Merk : Levis
Ukuran : L
Harga : 189500
Apakah anda akan menambahkan data baru ke stack (y/n)? n
Isi Stack :
Celana Biru Levis L 189500.0
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0

Data yang keluar : Celana Biru Levis L 189500.0
Elemen teratas : Kemeja Putih Styves XL 127000.0
Isi Stack :
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0
```

Pertanyaan :

1. Berapa banyak data pakaian yang dapat ditampung di dalam stack? Tunjukkan potongan kode program untuk mendukung jawaban Anda tersebut!

Jawab : Pada kelas Stack292, kapasitas stack ditentukan oleh nilai variabel size yang diinisialisasi saat pembuatan objek Stack292, yaitu yang diinisialisasi dengan size 5 (dapat menampung data hingga 5 pakaian). Berikut adalah baris programnya :

```
Stack292 stk = new Stack292(size:5);
```

2. Perhatikan class StackMain, pada saat memanggil fungsi push, parameter yang dikirimkan adalah p. Data apa yang tersimpan pada variabel p tersebut?

```
stk.push(p);
```

Jawab : Pada saat memanggil fungsi push() dalam kelas StackMain29, parameter yang dikirimkan adalah objek p yang merupakan instance dari kelas Pakaian29. Objek p akan berisi data pakaian yang dimasukkan oleh pengguna, yaitu jenis, warna, merk, ukuran, dan harga dari pakaian

3. Apakah fungsi penggunaan do-while yang terdapat pada class StackMain?

Jawab : Fungsi dari penggunaan do-while dalam kelas StackMain29 adalah untuk memungkinkan pengguna untuk terus menambahkan data pakaian ke dalam stack sampai pengguna memilih untuk berhenti menambahkan data.

4. Modifikasi kode program pada class StackMain sehingga pengguna dapat memilih operasi operasi pada stack (push, pop, peek, atau print) melalui pilihan menu program dengan memanfaatkan kondisi IF-ELSE atau SWITCH-CASE!

Jawab :

Berikut adalah kode program hasil modifikasinya :

```

1  import java.util.Scanner;
2
3  public class StackMain29 {
4      public static void main(String[] args) {
5          Scanner sc29 = new Scanner(System.in);
6          Stack292 stk = new Stack292(5);
7
8          char pilih;
9          do {
10             System.out.println("Menu:");
11             System.out.println("1. Push (Tambah data pakaian)");
12             System.out.println("2. Pop (Hapus data pakaian teratas)");
13             System.out.println("3. Peek (lihat data pakaian teratas)");
14             System.out.println("4. Print (Cetak semua data pakaian)");
15             System.out.println("5. Keluar");
16
17             System.out.print("Pilih operasi (1-5): ");
18             int choice = sc29.nextInt();
19             sc29.nextLine();
20
21             switch (choice) {
22                 case 1:
23                     System.out.print("Jenis : ");
24                     String jenis = sc29.nextLine();
25                     System.out.print("Warna : ");
26                     String warna = sc29.nextLine();
27                     System.out.print("Merk : ");
28                     String merk = sc29.nextLine();
29                     System.out.print("Ukuran : ");
30                     String ukuran = sc29.nextLine();
31                     System.out.print("Harga : ");
32                     double harga = sc29.nextDouble();
33                     sc29.nextLine();
34                     Pakaian29 p = new Pakaian29(jenis, warna, merk, ukuran, harga);
35                     stk.push(p);
36                     System.out.println("Data ditambahkan ke dalam stack.");
37                     break;
38                 case 2:
39                     stk.pop();
40                     System.out.println("Data pakaian teratas telah dihapus.");
41                     break;
42                 case 3:
43                     stk.peek();
44                     break;
45                 case 4:
46                     stk.print();
47                     break;
48                 case 5:
49                     System.out.println("Keluar dari program.");
50                     System.exit(0);
51                     break;
52                 default:
53                     System.out.println("Pilihan tidak valid. Silakan pilih 1-5.");
54             }
55
56             System.out.println("Isi stack setelah operasi:");
57             stk.print();
58
59             System.out.print("Apakah anda ingin melanjutkan (y/n)? ");
60             pilih = sc29.next().charAt(0);
61             sc29.nextLine();
62         } while (pilih == 'y' || pilih == 'Y');
63
64         sc29.close();
65     }
66 }
67

```

Output :

```
Menu:
1. Push (Tambah data pakaian)
2. Pop (Hapus data pakaian teratas)
3. Peek (Lihat data pakaian teratas)
4. Print (Cetak semua data pakaian)
5. Keluar
Pilih operasi (1-5): 1
Jenis : Kaos
Warna : Hitam
Merk : Nevada
Ukuran : M
Harga : 85000
Data ditambahkan ke dalam stack.
Isi stack setelah operasi:
Isi Stack :
Kaos Hitam Nevada M 85000.0

Apakah anda ingin melanjutkan (y/n)? y
Menu:
1. Push (Tambah data pakaian)
2. Pop (Hapus data pakaian teratas)
3. Peek (Lihat data pakaian teratas)
4. Print (Cetak semua data pakaian)
5. Keluar
Pilih operasi (1-5): 3
Elemen teratas : Kaos Hitam Nevada M 85000.0
Isi stack setelah operasi:
Isi Stack :
Kaos Hitam Nevada M 85000.0

Apakah anda ingin melanjutkan (y/n)?
```


PRAKTIKUM 3

Kode Program class PostFix29 :

```
1 public class PostFix29 {
2     int top;
3     char[] stack;
4
5     public PostFix29(int total) {
6         top = -1;
7         stack = new char[total];
8         push('/');
9     }
10
11     public void push(char c) {
12         top++;
13         stack[top] = c;
14     }
15
16     public char pop() {
17         char item = stack[top];
18         top--;
19         return item;
20     }
21
22     public boolean isOperand(char c) {
23         return (c == '+' || c == '-' || c == '*' || c == '/') || (c == '(' || c == ')');
24     }
25
26     public boolean isOperator(char c) {
27         return c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+';
28     }
29
30     public int derajat(char c) {
31         switch (c) {
32             case '^':
33                 return 3;
34             case '%':
35             case '/':
36             case '*':
37                 return 2;
38             case '-':
39             case '+':
40                 return 1;
41             default:
42                 return 0;
43         }
44     }
45
46     public String konversi(String Q) {
47         String P = "";
48         char c;
49         for (int i = 0; i < Q.length(); i++) {
50             c = Q.charAt(i);
51             if (isOperand(c)) {
52                 P += c;
53             } else if (c == '(') {
54                 push(c);
55             } else if (c == ')') {
56                 while (stack[top] != '(') {
57                     P += pop();
58                 }
59                 pop(); // for the '('
60             } else if (isOperator(c)) {
61                 while (top != -1 && derajat(stack[top]) >= derajat(c) && stack[top] != '(') {
62                     P += pop();
63                 }
64                 push(c);
65             }
66         }
67
68         while (top != -1) {
69             if (stack[top] == '(') {
70                 top--; // Remove the '('
71             } else {
72                 P += pop();
73             }
74         }
75         return P;
76     }
77 }
78 }
```

Kode Program class PostfixMain29 :

```
1 import java.util.Scanner;
2
3 public class PostfixMain29 {
4     public static void main(String[] args) {
5         Scanner sc29 = new Scanner(System.in);
6         String P, Q;
7         System.out.print("Masukkan ekspresi matematika (infix) : ");
8         Q = sc29.nextLine();
9         Q = Q.trim();
10        Q = Q + " ";
11
12        int total = Q.length();
13
14        Postfix29 post = new Postfix29(total);
15        P = post.konversi(Q);
16        System.out.println("Postfix : " + P);
17    }
18 }
19
```

Output :

```
Masukkan ekspresi matematika (infix) : a+b*(c+d-e)/f
Postfix : abcd+e-*f/+
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P9>
```

Pertanyaan :

1. Perhatikan class Postfix, jelaskan alur kerja method derajat!

Jawab : Di dalam kelas Postfix29, metode derajat(char c) digunakan untuk menentukan derajat prioritas operator matematika. Derajat prioritas digunakan untuk menentukan urutan operasi yang benar ketika mengonversi ekspresi infix menjadi postfix.

2. Apa fungsi kode program berikut?

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

Jawab : Kode program `c = Q.charAt(i);` memiliki fungsi untuk mengambil karakter pada indeks ke-i dari sebuah string Q dan menyimpannya ke dalam variabel c

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

Jawab : Berikut adalah hasilnya :

```
Masukkan ekspresi matematika (infix) : 5*4^(1+2)%3
Postfix : 5412+^*3%
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P9>
```

4. Pada soal nomor 3, mengapa tanda kurung tidak ditampilkan pada hasil konversi? Jelaskan!

Jawab : Karena tanda kurung bukan merupakan sebuah operator. Tanda "(" akan menge-pop top stack hingga bertemu tanda "("