

ALGORITMA STRUKTUR DATA

Praktikum – Stack

Lavina 2341760062

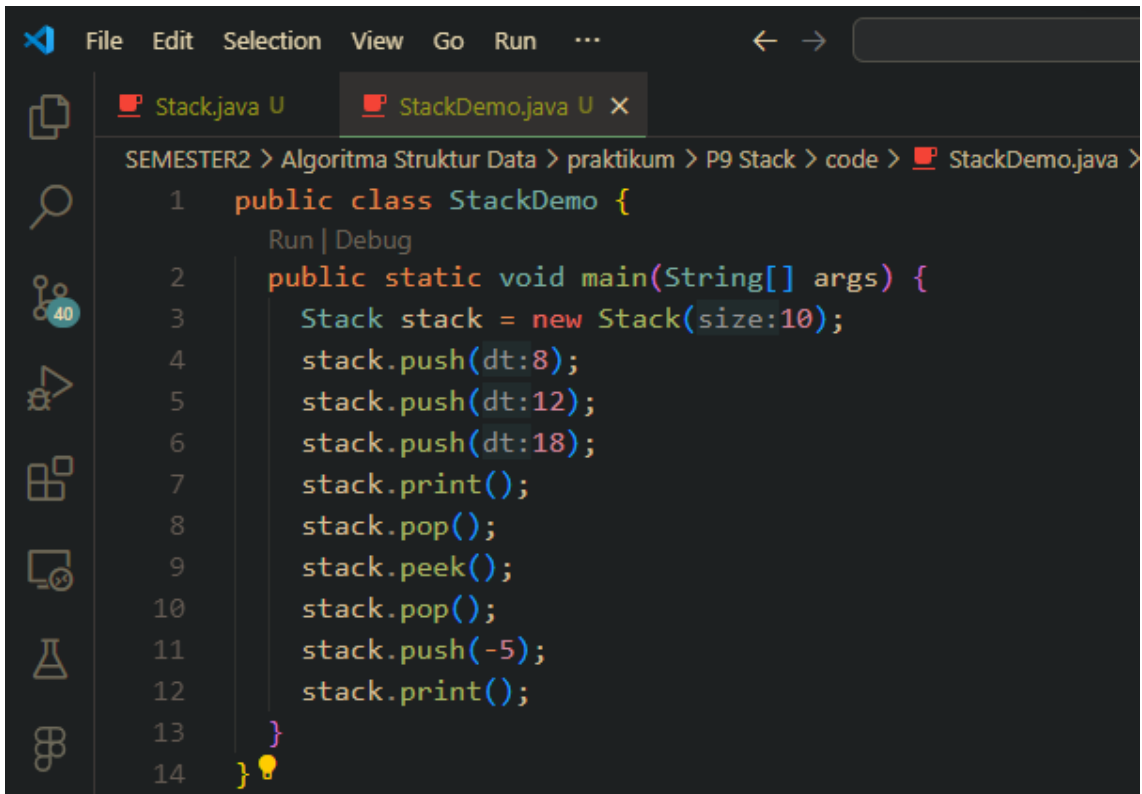
Praktikum 1

Stack.java

```
1  public class Stack {
2      int data[];
3      int size, top;
4
5      public Stack(int size) {
6          this.size = size;
7          data = new int[size];
8          top = -1;
9      }
10
11     public boolean isFull() {
12         if (top == size - 1) {
13             return true;
14         } else {
15             return false;
16         }
17     }
18
19     public boolean isEmpty() {
20         if (top == -1) {
21             return true;
22         } else {
23             return false;
24         }
25     }
26
27     public void push(int dt) {
28         if (!isFull()) {
29             top++;
30             data[top] = dt;
31         } else {
32             System.out.println("Stack Penuh");
33         }
34     }
35 }
```

```
36     public void pop() {
37         if (!isEmpty()) {
38             int x = data[top];
39             top--;
40             System.out.println("Data yang dikeluarkan dari stack: " + x);
41         } else {
42             System.out.println("Stack masih kosong");
43         }
44     }
45
46     public void peek() {
47         System.out.println("Elemen teratas stack: " + data[top]);
48     }
49
50     public void print() {
51         System.out.println("Isi stack: ");
52         for (int i = top; i >= 0; i--) {
53             System.out.println(data[i] + " ");
54         }
55         System.out.println("");
56     }
57
58     public void clear() {
59         if (!isEmpty()) {
60             for (int i = top; i >= 0; i--) {
61                 top--;
62             }
63             System.out.println("Stack sudah dikosongkan");
64         } else {
65             System.out.println("Stack masih kosong");
66         }
67     }
68 }
69
```

StackDemo.java



```
1 public class StackDemo {
2     Run | Debug
3     public static void main(String[] args) {
4         Stack stack = new Stack(size:10);
5         stack.push(dt:8);
6         stack.push(dt:12);
7         stack.push(dt:18);
8         stack.print();
9         stack.pop();
10        stack.peek();
11        stack.pop();
12        stack.push(-5);
13        stack.print();
14    }
```

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
```

Pertanyaan

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

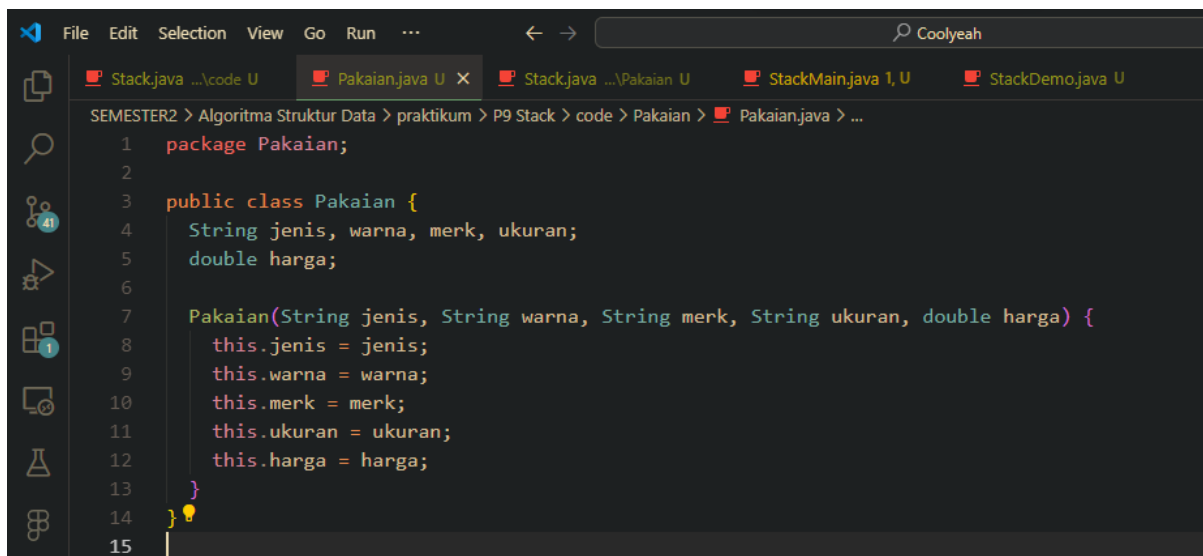
Jawab : Method tersebut dipanggil untuk mengecek apakah stacknya kosong atau tidak, karena jika stacknya kosong maka proses pop atau penghapusan data tidak bisa dilakukan.

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

Jawab : Perbedaannya adalah pada method peek() hanya menampilkan data yang ada dipaling atas stack tanpa menghapusnya sedangkan method pop() mengambil data teratas dari stack sehingga menghilangkan data tersebut juga.

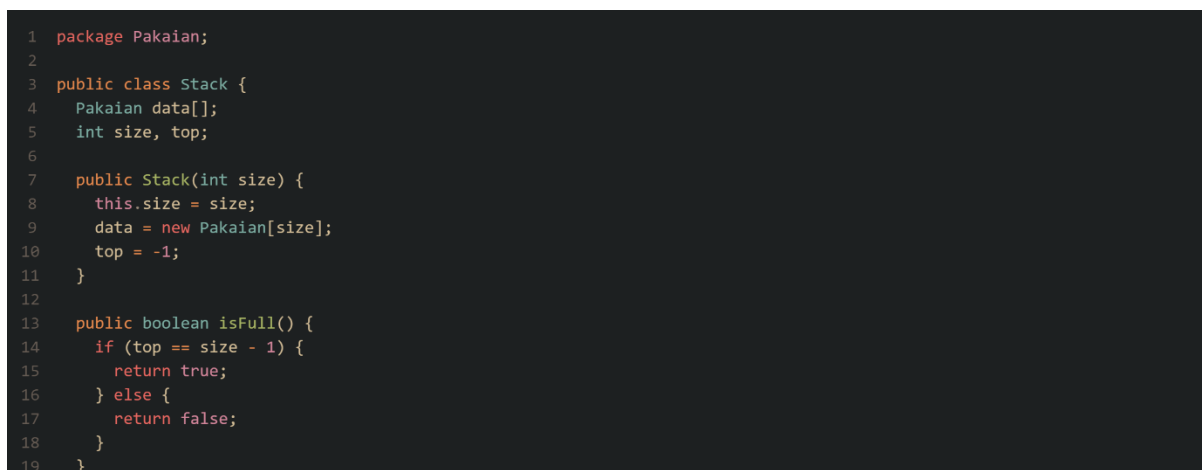
Percobaan 2

Pakaian.java



```
1 package Pakaian;
2
3 public class Pakaian {
4     String jenis, warna, merk, ukuran;
5     double harga;
6
7     Pakaian(String jenis, String warna, String merk, String ukuran, double harga) {
8         this.jenis = jenis;
9         this.warna = warna;
10        this.merk = merk;
11        this.ukuran = ukuran;
12        this.harga = harga;
13    }
14 }
15
```

Stack.java



```
1 package Pakaian;
2
3 public class Stack {
4     Pakaian data[];
5     int size, top;
6
7     public Stack(int size) {
8         this.size = size;
9         data = new Pakaian[size];
10        top = -1;
11    }
12
13    public boolean isFull() {
14        if (top == size - 1) {
15            return true;
16        } else {
17            return false;
18        }
19    }
20 }
```

```

21     public boolean isEmpty() {
22         if (top == -1) {
23             return true;
24         } else {
25             return false;
26         }
27     }
28
29     public void push(Pakaian pkn) {
30         if (!isFull()) {
31             top++;
32             data[top] = pkn;
33         } else {
34             System.out.println("Stack Penuh");
35         }
36     }
37
38     public void pop() {
39         if (!isEmpty()) {
40             Pakaian x = data[top];
41             top--;
42             System.out.println("Data yang dikeluarkan dari stack: " + x.jenis + " " + x.warna + " " + x.merk + " " + x.ukuran
43                 + " " + x.harga);
44         } else {
45             System.out.println("Stack masih kosong");
46         }
47     }
48
49     public void peek() {
50         System.out.println("Elemen teratas stack: " + data[top].jenis + " " + data[top].warna + " " + data[top].merk + " "
51             + data[top].ukuran + " " + data[top].harga);
52     }
53
54     public void print() {
55         System.out.println("Isi stack: ");
56         for (int i = top; i >= 0; i--) {
57             System.out.println(data[i].jenis + " " + data[i].warna + " " + data[i].merk + " "
58                 + data[i].ukuran + " " + data[i].harga);
59         }
60         System.out.println("");
61     }
62
63     public void clear() {
64         if (!isEmpty()) {
65             for (int i = top; i >= 0; i--) {
66                 top--;
67             }
68             System.out.println("Stack sudah dikosongkan");
69         } else {
70             System.out.println("Stack masih kosong");
71         }
72     }
73 }
74

```

StackMain.java

```

1  package Pakaian;
2
3  import java.util.Scanner;
4
5  public class StackMain {
6      public static void main(String[] args) {
7          Stack stk = new Stack(5);
8          Scanner sc = new Scanner(System.in);
9
10         char pilih;
11         do {

```

```

12     System.out.print("Jenis: ");
13     String jenis = sc.nextLine();
14     System.out.print("Warna: ");
15     String warna = sc.nextLine();
16     System.out.print("Merk: ");
17     String merk = sc.nextLine();
18     System.out.print("Ukuran: ");
19     String ukuran = sc.nextLine();
20     System.out.print("Harga: ");
21     double harga = sc.nextDouble();
22
23     Pakaian p = new Pakaian(jenis, warna, merk, ukuran, harga);
24     System.out.print("Apakah anda akan menambahkan data baru ke stack (y/n)? : ");
25     pilih = sc.next().charAt(0);
26     sc.nextLine();
27     stk.push(p);
28 } while (pilih == 'y');
29
30     stk.print();
31     stk.pop();
32     stk.peek();
33     stk.print();
34 }
35 }
36

```

Output :

```

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\Program Files\Java\jdk-21\bin\java.exe
1177a4bd3a26c42b680d90275\redhat.java\jdt_ws\Coolyeah_14a
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)? : y
Jenis: Hijab
Warna: Hijau
Merk: Umama
Ukuran: S
Harga: 50000
Apakah anda akan menambahkan data baru ke stack (y/n)? : n

```

```
Isi stack:
Hijab Hijau Umama S 50000.0
Celana Biru Levis L 189500.0
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0

Data yang dikeluarkan dari stack: Hijab Hijau Umama S 50000.0
Elemen teratas stack: Celana Biru Levis L 189500.0
Isi stack:
Celana Biru Levis L 189500.0
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 : Data yang bisa ditampung adalah 5, ditunjukkan pada sintaks **Stack stk = new Stack(5);**

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

Jawab : Variabel p berisi data jenis, warna, merk, ukuran, harga yang diinputkan melalui keyboard.

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

Jawab : Fungsinya agar bisa mengisi lebih dari 1 jenis pakaian, sehingga jika memilih "y" maka formnya akan diulang.

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!

```
1 package Pakaian;
2
3 import java.util.Scanner;
4
5 public class StackMain {
6     public static void main(String[] args) {
7         Stack stk = new Stack(5);
8         Scanner sc = new Scanner(System.in);
9
10        int menu;
11        do {
12            System.out.println("Menu");
13            System.out.println("1.Push\n2.Pop\n3.Peek\n4.Print\n5.Keluar");
14            System.out.print("Pilih: ");
15            menu = sc.nextInt();
16            sc.nextLine();
```

```

18     switch (menu) {
19         case 1:
20             char pilih;
21             do {
22                 System.out.print("Jenis: ");
23                 String jenis = sc.nextLine();
24                 System.out.print("Warna: ");
25                 String warna = sc.nextLine();
26                 System.out.print("Merk: ");
27                 String merk = sc.nextLine();
28                 System.out.print("Ukuran: ");
29                 String ukuran = sc.nextLine();
30                 System.out.print("Harga: ");
31                 double harga = sc.nextDouble();
32
33                 Pakaian p = new Pakaian(jenis, warna, merk, ukuran, harga);
34                 System.out.print("Apakah anda akan menambahkan data baru ke stack (y/n)? : ");
35                 pilih = sc.next().charAt(0);
36                 sc.nextLine();
37                 stk.push(p);
38             } while (pilih == 'y');
39             break;
40         case 2:
41             stk.pop();
42             break;
43         case 3:
44             stk.peek();
45             break;
46         case 4:
47             stk.print();
48             break;
49         case 5:
50             break;
51         default:
52             break;
53     }
54 } while (menu < 5);
55 }
56 }

```

Output :

```

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\Program Files\Java\jdk-21\bin\java
1177a4bd3a26c42b680d90275\redhat.java\jdt_ws\Coolyeah_14
Menu
1.Push
2.Pop
3.Peek
4.Print
5.Keluar
Pilih: 1
Jenis: Baju
Warna: Biru
Merk: HNM
Ukuran: XL
Harga: 80000
Apakah anda akan menambahkan data baru ke stack (y/n)? : N
Menu
1.Push
2.Pop
3.Peek
4.Print
5.Keluar
Pilih: 4
Isi stack:
Baju Biru HNM XL 80000.0

```



```

Menu
1.Push
2.Pop
3.Peek
4.Print
5.Keluar
Pilih: 3
Elemen teratas stack: Baju Biru HNM XL 80000.0
Menu
1.Push
2.Pop
3.Peek
4.Print
5.Keluar
Pilih: 2
Data yang dikeluarkan dari stack: Baju Biru HNM XL 80000.0
Menu
1.Push
2.Pop
3.Peek
4.Print
5.Keluar
Pilih: 4
Isi stack:

```

Percobaan 3

Postfix.java

```

1  package Postfix;
2
3  public class Postfix {
4      int n, top;
5      char[] stack;
6
7      public Postfix(int total) {
8          n = total;
9          top = -1;
10         stack = new char[n];
11         push('(');
12     }
13
14     public void push(char c) {
15         top++;
16         stack[top] = c;
17     }
18
19     public char pop() {
20         char item = stack[top];
21         top--;
22         return item;
23     }
24
25     public boolean IsOperand(char c) {
26         if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || c == ' ' || c == ',') {
27             return true;
28         } else {
29             return false;
30         }
31     }
32
33     public boolean IsOperator(char c) {
34         if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {
35             return true;
36         } else {
37             return false;
38         }
39     }

```

```

41 public int derajat(char c) {
42     switch (c) {
43         case '^':
44             return 3;
45         case '%':
46             return 2;
47         case '/':
48             return 2;
49         case '*':
50             return 2;
51         case '-':
52             return 1;
53         case '+':
54             return 1;
55         default:
56             return 0;
57     }
58 }
59
60 public String konversi(String Q) {
61     String P = "";
62     char c;
63
64     for (int i = 0; i < n; i++) {
65         c = Q.charAt(i);
66         if (IsOperand(c)) {
67             P = P + c;
68         }
69         if (c == '(') {
70             push(c);
71         }
72         if (c == ')') {
73             while (stack[top] != '(') {
74                 P = P + pop();
75             }
76             pop();
77         }
78         if (IsOperator(c)) {
79             while (derajat(stack[top]) >= derajat(c)) {
80                 P = P + pop();
81             }
82             push(c);
83         }
84     }
85     return P;
86 }
87 }

```

PostfixMain.java

```

1 package Postfix;
2
3 import java.util.Scanner;
4
5 public class PostfixMain {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         String P, Q;

```

```

10      System.out.println(x:"Masukkan ekspresi matematika (infix): ");
11      Q = sc.nextLine();
12      Q = Q.trim();
13      Q = Q + ")";
14
15      int total = Q.length();
16      Postfix post = new Postfix(total);
17      P = post.konversi(Q);
18      System.out.println("Postfix: " + P);
19  }
20  }

```

Output :

```

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\\Program\\ Files\\Java\\jdk-21\\bin
1177a4bd3a26c42b680d90275\\redhat.java\\jdt_ws\\Cooly
Masukkan ekspresi matematika (infix):
a+b*(c+d-e)/f
Postfix: abc*+d+ef

```

Pertanyaan

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

Jawab : Method ini berfungsi untuk menentukan tingkat atau operasi mana yang lebih tinggi untuk nantinya akan dioperasikan terlebih dahulu.

2. Apa fungsi kode program berikut?

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

Jawab : Untuk mengambil setiap karakter dari inputan atau memisahkan inputan dari pengguna menjadi array yang berisi karakter-karakter.

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

```

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)←[0m
$ /usr/bin/env C:\\Program\\ Files\\Java\\jdk-21\\bin\\ja
1177a4bd3a26c42b680d90275\\redhat.java\\jdt_ws\\Coolyeah_
Masukkan ekspresi matematika (infix):
5*4^(1+2)%3
Postfix: 541^+*23

```

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

Jawab : Karena tanda kurung hanya di-pop, tidak perlu dimasukkan ke postfix