

JOBSHEET 7

Nama : Farid Aziz Wicaksono

Kelas : TI/1C

Absen : 14

A. Praktikum

1. PRAKTIKUM 1 – Membuat Class Stack

```
1 package minggu7;
2
3 public class Stack {
4     int size;
5     int top;
6     int[] data;
7
8     public Stack(int size){
9         this.size = size;
10        data = new int[size];
11        top = -1;
12    }
13
14    public boolean IsEmpty(){
15        if(top==-1)
16            return true;
17        else
18            return false;
19    }
20
21    public boolean IsFull(){
22        if(top==size-1){
23            return true;
24        }
25        else{
26            return false;
27        }
28    }
29
30    public void push(int dt){
31        if(!IsFull()){
32            top++;
33            data[top]=dt;
34        }
35        else{
36
37            System.out.println("=====");
38            System.out.println("Isi Stack Penuh!");
39        }
```

```

40     }
41
42     public void pop(){
43         if(!IsEmpty()){
44             int a = data[top];
45             top--;
46
47             System.out.println("=====");
48             System.out.println("Data yang ada di Pop : " +a);
49         }
50         else{
51
52             System.out.println("=====");
53             System.out.println("Tidak dapat Pop; Stack masih kosong");
54         }
55     }
56
57     public void tampilkan(){
58         System.out.println("isi Stack : ");
59         for(int i = top; i>=0; i--){
60             System.out.println "["+data[i]+" ] ");
61         }
62     }
63
64     public void clearStack(){
65         if(!IsEmpty()){
66             for(int i = top; i>=0; i--){
67                 top--;
68             }
69
70
71             System.out.println("=====");
72             System.out.println("Stack Berhasil dikosongkan");
73         }
74         else{
75
76             System.out.println("=====");
77             System.out.println("Tidak dapat Hapus, Stack masih kosong");
78         }
79     }
80 }

```

TestStack

```
1 package minggu7;
2
3 public class TestStack {
4     public static void main(String[] args){
5         Stack tumpuk = new Stack(5);
6         tumpuk.push(20);
7         tumpuk.push(21);
8         tumpuk.push(12);
9         tumpuk.push(29);
10        tumpuk.push(11);
11
12        tumpuk.tampilkan();
13
14        tumpuk.pop();
15        tumpuk.tampilkan();
16    }
17 }
```

Output :

```
run:
isi Stack :
[11]
[29]
[12]
[21]
[20]
=====
Data yang ada di Pop : 11
isi Stack :
[29]
[12]
[21]
[20]
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. PRAKTIKUM 2 – Membuat Class Postfix

```
1 package minggu7;
2
3 public class Postfix {
4     int n, top;
5     char[] stack;
6
7     Postfix(int total){
8         n = total;
9         top = -1;
10        stack = new char[n];
11        push('(');
12    }
13
14    void push(char c){
15        top++;
16        stack[top]=c;
17    }
18
19    char pop(){
20        char item = stack[top];
21        top--;
22        return item;
23    }
24
25    String convert(String Q){
26        String P = "";
27        char c;
28        for(int i=0; i<n; i++){
29            c = Q.charAt(i);
30            if(isOperand(c))
31                P=P+c;
32            if(c=='(')
33                push(c);
34            if(isOperator(c)){
35                while(presedence(stack[top])>=presedence(c))
36                    P =P+pop();
37                push(c);
38            }
39            if(c==')'){
40                while(stack[top]!='(')
41                    P = P + pop();
42                char temp = pop();
43            }
44        }
45        return P;
46    }
47 }
```

```

44     }
45
46     int presedence(char c){
47         switch(c){
48             case '^' : return 3;
49             case '%' : return 2;
50             case '/' : return 2;
51             case '*' : return 2;
52             case '-' : return 1;
53             case '+' : return 1;
54             default : return 0;
55         }
56     }
57
58     boolean isOperand(char c){
59         if ((c>='A' && c<='Z') || (c>='a' && c<='z') || (c>='0' && c<='9') ||
60             c=='.' || c=='.')
61             return true;
62         else
63             return false;
64     }
65
66     boolean isOperator(char c){
67         if(c=='^' || c=='%' || c=='/' || c=='*' || c=='-' || c=='+')
68             return true;
69         else
70             return false;
71     }
72 }

```

Class main

```

1  package minggu7;
2  import java.util.Scanner;
3  public class main {
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6          String P,Q;
7
8          System.out.println("Masukan Ekspresi :");
9          Q = sc.nextLine();
10         Q = Q.trim();
11         Q = Q + " ";
12
13         int total = Q.length();
14         Postfix pf = new Postfix(total);

```

15	P = pf.convert(Q);
16	System.out.println("Postfix:\n"+P);
17	}
18	}

Output :

```
run:
Masukan Ekspresi :
a*(b+c)
Postfix:
abc+*
BUILD SUCCESSFUL (total time: 13 seconds)
```

B. Pertanyaan

1. Lakukan penambahan data pada program praktikum 1 di atas yaitu angka 21 dan 45, dan tampilkan datanya!
2. Pada praktikum 1 langkah pertama yang sudah anda lakukan mengapa data yang bisa dimasukkan didalam stack hanya angka 21 sedangkan angka 45 tidak bisa, jelaskan!
3. Modifikasi program pratikum 1 diatas dengan memanggil method pop() kembali, kemudian masukkan angka 45 di dalam stack, selanjutya cetaklah datanya juga, amati apakah yang terjadi dan jelaskan!
4. Apakah fungsi method pop() pada program diatas?
5. Modifikasilah program praktikum 1 di atas menggunakan inputan Scanner!
6. Mengapa variabel top diatas diinisialisasi degan nilai "-1" pada 2 praktikum di atas?

```
int size;
int top=-1;
int[] data;
atau
PostFix(int total) {
    n = total;
    top = -1;
    stack = new char[n];
    push('(');
}
```

7. Apakah fungsi pembuatan method presedence pada praktikum 2!

8. Setelah inputan ekspresi bertipe string pada praktikum 2 dilakukan pemanggilan fungsi trim(). Apakah kegunaanya?
- ```
String P,Q;
System.out.println("Masukkan Ekspresi :");
Q = sc.nextLine();
Q = Q.trim();
```
9. Pada praktikum 2, jika pada ekspresi inputan terdapat tanda kurung, maka akan otomatis dihilangkan. Bagaimanakah caranya?terdapat pada method bagian mana cara tersebut?
10. Apabila diinputkan ekspresi selain operand dan operator pada praktikum 2. Misalkan saja tanda "@" atau "#". Apakah output yang dihasilkan?Jelaskan!

Jawab :

```
1. public class TestStack {
 public static void main(String[] args){
 Stack tumpuk = new Stack(5);
 tumpuk.push(20);
 tumpuk.push(21);
 tumpuk.push(12);
 tumpuk.push(29);
 tumpuk.push(11);
 tumpuk.tampilkan();
 tumpuk.pop();
 tumpuk.tampilkan();
 tumpuk.push(21);
 tumpuk.push(45);
 tumpuk.tampilkan();
 }
}
```

2. karena isi dalam stack telah penuh setelah terisi angka 21 (batas isi stack 5).

```
3. public class TestStack {
 public static void main(String[] args){
 Stack tumpuk = new Stack(5);

 tumpuk.push(20);
 tumpuk.push(21);
 tumpuk.push(12);
 tumpuk.push(29);
 tumpuk.push(11);
 tumpuk.tampilkan();
 tumpuk.pop();
```

```

 tumpuk.tampilkan();
 tumpuk.push(21);
 tumpuk.tampilkan();
 tumpuk.pop();
 tumpuk.push(45);
 tumpuk.tampilkan();
 }}

```

4. untuk mengambil item pada stack pada tumpukan paling atas.

```

5. import java.util.Scanner;
 public class TestStack {
 public static void main(String[] args){
 Scanner sc = new Scanner(System.in);
 int N, k;

 System.out.print("Masukan jumlah Stack : ");
 N = sc.nextInt();
 Stack tumpuk = new Stack(N);

 for(int i = 0; i<N; i++){
 System.out.print("Masukan angka : ");
 k = sc.nextInt();
 tumpuk.push(k);
 }
 tumpuk.tampilkan();
 }
 }

```

6. Pada mulanya isi top dengan -1, karena array dalam dimulai dari 0, yang berarti bahwa data stack adalah kosong!

7. Untuk menentukan derajat suatu operator.

8. Pemotongan per karakter

```

9. if(c==''){
 while(stack[top]!='(')
 p = p + pop();
 char temp = pop();

```

Karena apabila tanda “(“ bertemu dengan “)” akan hilang

10. Operator @ dan # tidak akan muncul karena tidak ada di method isOperand / isOperator.



### C. Tugas

1. Modifikasilah program pada praktikum yang sudah anda lakukan yang inputannya berupa angka gantilah menjadi data kalimat.
2. Buatlah program dengan menggunakan konsep stack untuk menginputkan sebuah kalimat kemudian outputnya adalah berupa kalimat terbalik.

```
run:
Masukkan Kalimat : polinema jaya
Kalimat dibalik : jaya polinema
BUILD SUCCESSFUL (total time: 7 seconds)
```

3. Buatlah suatu class sesuai praktikum 2 untuk mengembalikan notasi postfix menjadi infix!
4. Apakah kegunaan notasi prefix dalam kehidupan sehari?
5. Buatlah slass untuk mengkonversi infix menjadi prefix!

Jawab :

#### 1. Stack

```
1 package minggu7;
2
3 public class Stack {
4 int size;
5 int top;
6 String[] data;
7
8 public Stack(int size){
9 this.size = size;
10 data = new String[size];
11 top = -1;
12 }
13
14 public boolean IsEmpty(){
15 if(top== -1)
16 return true;
17 else
18 return false;
19 }
20
21 public boolean IsFull(){
22 if(top==size-1){
23 return true;
24 }
25 else{
26 return false;
```

```

27 }
28 }
29
30 public void push(String dt){
31 if(!IsFull()){
32 top++;
33 data[top]=dt;
34 }
35 else{
36
37 System.out.println("=====");
38 System.out.println("Isi Stack Penuh!");
39 }
40 }
41
42 public void pop(){
43 if(!IsEmpty()){
44 String a = data[top];
45 top--;
46
47 System.out.println("=====");
48 System.out.println("Data yang ada di Pop : " +a);
49 }
50 else{
51
52 System.out.println("=====");
53 System.out.println("Tidak dapat Pop; Stack masih kosong");
54 }
55 }
56
57 public void tampilkan(){
58 System.out.println("isi Stack : ");
59 for(int i = top; i>=0; i--){
60 System.out.println "["+data[i]+" ");
61 }
62 }
63
64 public void clearStack(){
65 if(!IsEmpty()){
66 for(int i = top; i>=0; i--){
67 top--;
68 }
69
70
71 System.out.println("=====");
72 System.out.println("Stack Berhasil dikosongkan");

```

```

73 }
74 else{
75
76 System.out.println("=====");
77 System.out.println("Tidak dapat Hapus, Stack masih kosong");
78 }
79 }
80 }

```

#### TestStack

```

1 package minggu7;
2
3 public class TestStack {
4 public static void main(String[] args){
5 Stack tumpuk = new Stack(5);
6 tumpuk.push("goreng");
7 tumpuk.push("nasi");
8 tumpuk.push("makan");
9 tumpuk.push("suka");
10 tumpuk.push("saya");
11
12 tumpuk.tampilkan();
13
14 tumpuk.pop();
15 tumpuk.tampilkan();
16 }
17 }

```

#### Output :

```

run:
isi Stack :
[saya]
[suka]
[makan]
[nasi]
[goreng]
=====
Data yang ada di Pop : saya
isi Stack :
[suka]
[makan]
[nasi]
[goreng]
BUILD SUCCESSFUL (total time: 0 seconds)

```

## 2. Stack

```
1 package minggu7;
2
3 public class Stack {
4 int size;
5 int top;
6 String[] data;
7
8 public Stack(int size){
9 this.size = size;
10 data = new String[size];
11 top = -1;
12 }
13
14 public boolean IsEmpty(){
15 if(top== -1)
16 return true;
17 else
18 return false;
19 }
20
21 public boolean IsFull(){
22 if(top==size-1){
23 return true;
24 }
25 else{
26 return false;
27 }
28 }
29
30 public void push(String dt){
31 if(!IsFull()){
32 top++;
33 data[top]=dt;
34 }
35 else{
36
37 System.out.println("=====");
38 System.out.println("Isi Stack Penuh!");
39 }
40 }
41
42 public void pop(){
43 if(!IsEmpty()){
44 String a = data[top];
45 top--;
```

```

46
47 System.out.println("=====");
48 System.out.println("Data yang ada di Pop : " +a);
49 }
50 else{
51
52 System.out.println("=====");
53 System.out.println("Tidak dapat Pop; Stack masih kosong");
54 }
55 }
56
57 public void tampilkan(){
58 System.out.println("isi Stack : ");
59 for(int i = top; i>=0; i--){
60 System.out.println "["+data[i]+" ");
61 }
62 }
63
64 public void clearStack(){
65 if(!IsEmpty()){
66 for(int i = top; i>=0; i--){
67 top--;
68 }
69
70
71 System.out.println("=====");
72 System.out.println("Stack Berhasil dikosongkan");
73 }
74 else{
75
76 System.out.println("=====");
77 System.out.println("Tidak dapat Hapus, Stack masih kosong");
78 }
79 }
80 }

```

#### TestStack

```

1 package minggu7;
2 import java.util.Scanner;
3 public class TestStack {
4 public static void main(String[] args){
5 Scanner sc = new Scanner(System.in);
6 Scanner sa = new Scanner(System.in);
7
8 System.out.print("Tentukan isi stack : ");
9 int n = sc.nextInt();

```

|    |                                         |
|----|-----------------------------------------|
| 10 |                                         |
| 11 | Stack d = new Stack(n);                 |
| 12 | for(int i=0; i<n; i++){                 |
| 13 | System.out.print("Masukan Kalimat : "); |
| 14 | String k = sa.nextLine();               |
| 15 | d.push(k);                              |
| 16 | }                                       |
| 17 | d.tampilkan();                          |
| 18 | }                                       |
| 19 | }                                       |

Output :

```
run:
Tentukan isi stack : 2
Masukan Kalimat : polinema
Masukan Kalimat : jaya
isi Stack :
[jaya]
[polinema]
BUILD SUCCESSFUL (total time: 19 seconds)
```

- 3.
4. Untuk mempermudah cara kerja program computer
- 5.

