

**LAPORAN PRAKTIKUM
ALGORITMA & STRUKTUR DATA**

Pertemuan – 9: Stack

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



MARGA RETA NOVIA PUTRI

2341760017

D-1V SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

7.2. Praktikum 1

Kode program Stack19.java :

```
1 public class Stack19 {
2     int data[];
3     int size;
4     int top;
5
6     public Stack19(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 di kosongkan");
65        } else {
66            System.out.println("Stack masih kosong");
67        }
68    }
69 }
```

Kode program StackDemo19.java :

```
1  public class StackDemo19 {  
2      public class StackDemo14 {  
3          public static void main(String[] args) {  
4              Stack19 stack = new Stack19(10);  
5              stack.push(8);  
6              stack.push(12);  
7              stack.push(18);  
8              stack.print();  
9              stack.pop();  
10             stack.peek();  
11             stack.pop();  
12             stack.push(-5);  
13             stack.print();  
14         }  
15     }  
16 }  
17
```

Hasil run kode program:

```
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
```

7.2.3 Pertanyaan

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

Jawab: Untuk memastikan bahwa stack tidak kosong sebelum mencoba untuk mengeluarkan elemen dari stack. Jika tidak ada pemanggilan `isEmpty()` sebelum operasi `pop()`, maka ada risiko mencoba mengakses indeks array yang tidak valid jika stack dalam keadaan kosong (indeks top akan bernilai -1).

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

Jawab: Method peek() digunakan untuk melihat nilai elemen teratas (atau puncak) dari stack tanpa menghapusnya dari stack sedangkan **Method pop()** digunakan untuk menghapus dan mengembalikan nilai dari elemen teratas (atau puncak) dari stack

7.3. Praktikum 2

Kode program Pakaian19.java :

```
1 public class Pakaian19 {
2
3     String jenis, warna, merk, ukuran;
4     double harga;
5
6     Pakaian19 (String jenis, String warna, String merk, String ukuran, double harga) {
7         this.jenis = jenis;
8         this.warna = warna;
9         this.merk = merk;
10        this.ukuran = ukuran;
11        this.harga = harga;
12    }
13 }
14
```

Kode program StackPakaian19.java :

```
1 public class StackPakaian19 {
2     int size;
3     int top;
4     Pakaian19 data[];
5
6     public StackPakaian19 (int size) {
7         this.size = size;
8         data = new Pakaian19[size];
9         top = -1;
10    }
11
12    public boolean isEmpty() {
13        if (top == -1) {
14            return true;
15        } else {
16            return false;
17        }
18    }
19
20    public boolean isFull() {
21        if (top == size -1) {
22            return true;
23        } else {
24            return false;
25        }
26    }
27
28    public void push(Pakaian19 pkn19) {
29        if (!isFull()) {
30            top++;
31            data[top] = pkn19;
32        } else {
33            System.out.println("Isi Stack penuh!!");
34        }
35    }
36
37    public void pop(){
38        if (!isEmpty()){
39            Pakaian19 x = data[top];
40            top--;
41            System.out.println("Data yang keluar: " + x.jenis + " " + x.warna + " " + x.merk + " " + x.ukuran + " " + x.harga);
42        } else {
43            System.out.println("Stack masih kosong");
44        }
45    }
46
47    public void peek() {
48        System.out.println("Elemen teratas: " + data[top].jenis + " " + data[top].warna + " " + data[top].merk + " " +
49            data[top].ukuran + " " + data[top].harga);
50    }
51
52    public void print() {
53        System.out.println("Isi stack: ");
54        for (int i = top; i >= 0; i--) {
55            System.out.println(data[i].jenis + " " + data[i].warna + " " + data[i].merk + " " + data[i].ukuran + " " + data[i].harga + " ");
56        }
57        System.out.println("");
58    }
59
60    public void clear() {
61        if (!isEmpty()) {
62            for (int i = top; i >= 0; i--) {
63                top--;
64            }
65            System.out.println("Stack sudah dikosongkan");
66        } else {
67            System.out.println("Stack masih kosong");
68        }
69    }
70 }
71
```

Kode program StackMain19.java :

```
1  import java.util.Scanner;
2  import java.util.Stack;
3
4  public class StackMain19 {
5      public static void main(String[] args) {
6
7          StackPakaian19 stk19 = new StackPakaian19(5);
8          Scanner sc19 = new Scanner(System.in);
9
10         char pilih;
11
12         do{
13             System.out.print("Jenis: ");
14             String jenis = sc19.nextLine();
15             System.out.print("Warna: ");
16             String warna = sc19.nextLine();
17             System.out.print("Merk: ");
18             String merk = sc19.nextLine();
19             System.out.print("Ukuran: ");
20             String ukuran = sc19.nextLine();
21             System.out.print("Harga: ");
22             double harga = sc19.nextDouble();
23
24             Pakaian19 p19 = new Pakaian19(jenis, warna, merk, ukuran, harga);
25             System.out.print("Apakah anda akan menambahkan data baru ke stack (y/n)? ");
26             pilih = sc19.next().charAt(0);
27             sc19.nextLine();
28             stk19.push(p19);
29         } while (pilih == 'y');
30
31         stk19.print();
32         stk19.pop();
33         stk19.peek();
34         stk19.print();
35     }
36 }
```

Hasil run kode program:

```
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
```

7.3.3. Pertanyaan

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

Jawab: Data pakaian yang dapat ditampung di dalam kelas StackPakaian19 adalah **5**. Dalam kelas **StackPakaian19** kapasitas stack ditentukan oleh nilai size yang diberikan pada konstruktor saat membuat objek:

```
7 StackPakaian19 stk19 = new StackPakaian19(size: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 menyimpan data spesifik tentang pakaian yang baru saja dimasukkan oleh pengguna, seperti jenis, warna, merk, ukuran, dan harga.

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

Jawab: Digunakan untuk melakukan iterasi untuk memasukkan data pakaian ke dalam stack, user dapat terus memasukkan data pakaian ke dalam stack sampai user memilih untuk berhenti dengan menjawab 'n' pada pertanyaan "Apakah anda akan menambahkan data baru ke stack (y/n)?".

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:

Kode program:

```
29 } while (pilih == 'y');
30
31 int input;
32 boolean lanjut = true;
33
34 while (lanjut) {
35     System.out.println(x:"\nPilih operasi stack yang di inginkan: ");
36     System.out.println(x:"1. Fungsi Pop");
37     System.out.println(x:"2. Fungsi Peek");
38     System.out.println(x:"3. Print");
39     System.out.println(x:"4. Keluar");
40     System.out.print(s:"Masukkan pilihan anda: ");
41     input = sc19.nextInt();
42
43     switch (input) {
44         case 1:
45             stk19.pop();
46             break;
47         case 2:
48             stk19.peek();
49             break;
50         case 3:
51             stk19.print();
52             break;
53         case 4:
54             lanjut = false;
55             break;
56         default:
57             System.out.println(x:"Pilihan anda salah harap ulangi lagi");
58     }
59 }
60
61 System.out.println(x:"Program selesai.");
62 sc19.close();
```

Hasil run kode program:

```
Pilih operasi stack yang di inginkan:
1. Fungsi Pop
2. Fungsi Peek
3. Print
4. Keluar
Masukkan pilihan anda: 1
Data yang keluar: Celana Biru Levis L 189500.0

Pilih operasi stack yang di inginkan:
1. Fungsi Pop
2. Fungsi Peek
3. Print
4. Keluar
Masukkan pilihan anda: 2
Elemen teratas: Kemeja Putih Styves XL 127000.0

Pilih operasi stack yang di inginkan:
1. Fungsi Pop
2. Fungsi Peek
3. Print
4. Keluar
Masukkan pilihan anda: 3
Isi stack:
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0

Pilih operasi stack yang di inginkan:
1. Fungsi Pop
2. Fungsi Peek
3. Print
4. Keluar
Masukkan pilihan anda: 4
Program selesai.
PS D:\SEMESTER2\ASD\PRAKTIKUM\jobsheet9> █
```


7.4. Praktikum 3

Kode program Postfix19.java :

```
1 import java.util.Scanner;
2
3 public class Postfix19 {
4
5     int n, top;
6     char[] stack;
7
8     public Postfix19 (int total) {
9         n = total;
10        top = -1;
11        stack = new char[n];
12        push('(');
13    }
14
15    private void push(char c) {
16        top++;
17        stack[top] = c;
18    }
19
20    public char pop() {
21        char item = stack[top];
22        top--;
23        return item;
24    }
25
26    public boolean IsOperand (char c) {
27        if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || c == '.' || c == '-') {
28            return true;
29        } else {
30            return false;
31        }
32    }
33
34    public boolean IsOperator(char c) {
35        if (c == '^' || c == 'x' || c == '/' || c == '*' || c == '-' || c == '+') {
36            return true;
37        } else {
38            return false;
39        }
40    }
41
42    public int derajat (char c) {
43        switch (c) {
44            case '^':
45                return 3;
46            case 'x':
47                return 2;
48            case '/':
49                return 2;
50            case '+':
51                return 2;
52            case '-':
53                return 1;
54            case '*':
55                return 1;
56            default:
57                return 0;
58        }
59    }
60
61    public String konversi(String Q) {
62        String P = "";
63        char c;
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 (top >= 0 && derajat(stack[top]) >= derajat(c)) {
80                    P = P + pop();
81                }
82                push(c);
83            }
84        }
85        return P;
86    }
87 }
```

Kode program PostfixMain19.java :

```
1  import java.util.Scanner;
2
3  public class PosfixMain19 {
4      public static void main(String[] args) {
5          Scanner sc19 = new Scanner(System.in);
6          String P, Q;
7          System.out.println("Masukkan ekspresi matematika (infix): ");
8          Q = sc19.nextLine();
9          Q = Q.trim();
10         Q = Q + " ";
11
12         int total = Q.length();
13
14         Postfix19 post = new Postfix19(total);
15         P = post.konversi(Q);
16         System.out.println("Postfix: " + P);
17     }
18 }
```

Hasil run kode program:

```
Masukkan ekspresi matematika (infix):
a+b*(c+d-e)/f
Postfix: abcd+e-*f/+
```

7.4.3. Pertanyaan

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

Jawab: Method derajat pada class postfix digunakan untuk menentukan prioritas operator dalam ekspresi matematika. Ini dilakukan dengan menggunakan switch statement yang mengevaluasi karakter operator c. Setiap operator memiliki prioritas yang berbeda, diwakili oleh nilai yang dikembalikan oleh method derajat.

2. Apa fungsi kode program berikut?

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

Jawab: Untuk mengambil karakter pada posisi tertentu dalam string Q dan menyimpannya ke dalam variabel c.

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

Jawab:

```
Masukkan ekspresi matematika (infix):  
5*4^(1+2)%3  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -1 out of bounds for length 12  
    at Postfix19.konversi(Postfix19.java:73)  
    at PosfixMain19.main(PosfixMain19.java:15)
```

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

Jawab: Karena tanda kurung tidak dimasukkan ke dalam ekspresi postfix.

7.5. Tugas

1. Perhatikan dan gunakan kembali kode program pada Praktikum 2. Tambahkan method getMax pada class Stack yang digunakan untuk mencari dan menampilkan data pakaian dengan harga tertinggi dari semua data pakaian yang tersimpan di dalam stack!

Jawab:

Kode program StackPakaian19.java:

```

1 public class StackRandomly {
2     int size;
3     int top;
4     Random rand = new Random();
5
6     public StackRandomly (int size) {
7         this.size = size;
8         data = new Randomly[size];
9         top = -1;
10    }
11
12    public boolean isEmpty() {
13        if (top == -1) {
14            return true;
15        }
16        return false;
17    }
18
19    public boolean isFull() {
20        if (top == size - 1) {
21            return true;
22        }
23        return false;
24    }
25
26    public void push(Object obj) {
27        if (isFull()) {
28            System.out.println("Stack penuh!");
29        }
30    }
31
32    public void pop() {
33        if (isEmpty()) {
34            System.out.println("Stack kosong!");
35        }
36        data[top] = null;
37        top--;
38        System.out.println("Stack sudah kosong!");
39    }
40
41    public void peek() {
42        System.out.println("Elemen terakhir : " + data[top]);
43        System.out.println("Elemen ke-2 : " + data[top-1]);
44        System.out.println("Elemen ke-3 : " + data[top-2]);
45        System.out.println("Elemen ke-4 : " + data[top-3]);
46        System.out.println("Elemen ke-5 : " + data[top-4]);
47    }
48
49    public void print() {
50        System.out.println("Stack :");
51        for (int i = 0; i < data.length; i++) {
52            System.out.print(data[i]);
53            System.out.print(" ");
54        }
55        System.out.println("");
56    }
57
58    public void clear() {
59        if (isEmpty()) {
60            for (int i = 0; i < data.length; i++) {
61                data[i] = null;
62            }
63            System.out.println("Stack sudah dihapus!");
64        }
65    }
66
67    public void generate() {
68        if (isEmpty()) {
69            System.out.println("Stack kosong!");
70        }
71        for (int i = 0; i < data.length; i++) {
72            data[i] = rand.nextInt(100);
73        }
74        System.out.println("Stack sudah terisi dengan angka 0-100!");
75        System.out.println("Elemen ke-1 : " + data[0]);
76        System.out.println("Elemen ke-2 : " + data[1]);
77        System.out.println("Elemen ke-3 : " + data[2]);
78        System.out.println("Elemen ke-4 : " + data[3]);
79        System.out.println("Elemen ke-5 : " + data[4]);
80        System.out.println("Elemen ke-6 : " + data[5]);
81        System.out.println("Elemen ke-7 : " + data[6]);
82        System.out.println("Elemen ke-8 : " + data[7]);
83        System.out.println("Elemen ke-9 : " + data[8]);
84        System.out.println("Elemen ke-10 : " + data[9]);
85        System.out.println("Elemen ke-11 : " + data[10]);
86        System.out.println("Elemen ke-12 : " + data[11]);
87        System.out.println("Elemen ke-13 : " + data[12]);
88        System.out.println("Elemen ke-14 : " + data[13]);
89        System.out.println("Elemen ke-15 : " + data[14]);
90        System.out.println("Elemen ke-16 : " + data[15]);
91        System.out.println("Elemen ke-17 : " + data[16]);
92        System.out.println("Elemen ke-18 : " + data[17]);
93        System.out.println("Elemen ke-19 : " + data[18]);
94        System.out.println("Elemen ke-20 : " + data[19]);
95        System.out.println("Elemen ke-21 : " + data[20]);
96        System.out.println("Elemen ke-22 : " + data[21]);
97        System.out.println("Elemen ke-23 : " + data[22]);
98        System.out.println("Elemen ke-24 : " + data[23]);
99        System.out.println("Elemen ke-25 : " + data[24]);
100       System.out.println("Elemen ke-26 : " + data[25]);
101       System.out.println("Elemen ke-27 : " + data[26]);
102       System.out.println("Elemen ke-28 : " + data[27]);
103       System.out.println("Elemen ke-29 : " + data[28]);
104       System.out.println("Elemen ke-30 : " + data[29]);
105       System.out.println("Elemen ke-31 : " + data[30]);
106       System.out.println("Elemen ke-32 : " + data[31]);
107       System.out.println("Elemen ke-33 : " + data[32]);
108       System.out.println("Elemen ke-34 : " + data[33]);
109       System.out.println("Elemen ke-35 : " + data[34]);
110       System.out.println("Elemen ke-36 : " + data[35]);
111       System.out.println("Elemen ke-37 : " + data[36]);
112       System.out.println("Elemen ke-38 : " + data[37]);
113       System.out.println("Elemen ke-39 : " + data[38]);
114       System.out.println("Elemen ke-40 : " + data[39]);
115       System.out.println("Elemen ke-41 : " + data[40]);
116       System.out.println("Elemen ke-42 : " + data[41]);
117       System.out.println("Elemen ke-43 : " + data[42]);
118       System.out.println("Elemen ke-44 : " + data[43]);
119       System.out.println("Elemen ke-45 : " + data[44]);
120       System.out.println("Elemen ke-46 : " + data[45]);
121       System.out.println("Elemen ke-47 : " + data[46]);
122       System.out.println("Elemen ke-48 : " + data[47]);
123       System.out.println("Elemen ke-49 : " + data[48]);
124       System.out.println("Elemen ke-50 : " + data[49]);
125       System.out.println("Elemen ke-51 : " + data[50]);
126       System.out.println("Elemen ke-52 : " + data[51]);
127       System.out.println("Elemen ke-53 : " + data[52]);
128       System.out.println("Elemen ke-54 : " + data[53]);
129       System.out.println("Elemen ke-55 : " + data[54]);
130       System.out.println("Elemen ke-56 : " + data[55]);
131       System.out.println("Elemen ke-57 : " + data[56]);
132       System.out.println("Elemen ke-58 : " + data[57]);
133       System.out.println("Elemen ke-59 : " + data[58]);
134       System.out.println("Elemen ke-60 : " + data[59]);
135       System.out.println("Elemen ke-61 : " + data[60]);
136       System.out.println("Elemen ke-62 : " + data[61]);
137       System.out.println("Elemen ke-63 : " + data[62]);
138       System.out.println("Elemen ke-64 : " + data[63]);
139       System.out.println("Elemen ke-65 : " + data[64]);
140       System.out.println("Elemen ke-66 : " + data[65]);
141       System.out.println("Elemen ke-67 : " + data[66]);
142       System.out.println("Elemen ke-68 : " + data[67]);
143       System.out.println("Elemen ke-69 : " + data[68]);
144       System.out.println("Elemen ke-70 : " + data[69]);
145       System.out.println("Elemen ke-71 : " + data[70]);
146       System.out.println("Elemen ke-72 : " + data[71]);
147       System.out.println("Elemen ke-73 : " + data[72]);
148       System.out.println("Elemen ke-74 : " + data[73]);
149       System.out.println("Elemen ke-75 : " + data[74]);
150       System.out.println("Elemen ke-76 : " + data[75]);
151       System.out.println("Elemen ke-77 : " + data[76]);
152       System.out.println("Elemen ke-78 : " + data[77]);
153       System.out.println("Elemen ke-79 : " + data[78]);
154       System.out.println("Elemen ke-80 : " + data[79]);
155       System.out.println("Elemen ke-81 : " + data[80]);
156       System.out.println("Elemen ke-82 : " + data[81]);
157       System.out.println("Elemen ke-83 : " + data[82]);
158       System.out.println("Elemen ke-84 : " + data[83]);
159       System.out.println("Elemen ke-85 : " + data[84]);
160       System.out.println("Elemen ke-86 : " + data[85]);
161       System.out.println("Elemen ke-87 : " + data[86]);
162       System.out.println("Elemen ke-88 : " + data[87]);
163       System.out.println("Elemen ke-89 : " + data[88]);
164       System.out.println("Elemen ke-90 : " + data[89]);
165       System.out.println("Elemen ke-91 : " + data[90]);
166       System.out.println("Elemen ke-92 : " + data[91]);
167       System.out.println("Elemen ke-93 : " + data[92]);
168       System.out.println("Elemen ke-94 : " + data[93]);
169       System.out.println("Elemen ke-95 : " + data[94]);
170       System.out.println("Elemen ke-96 : " + data[95]);
171       System.out.println("Elemen ke-97 : " + data[96]);
172       System.out.println("Elemen ke-98 : " + data[97]);
173       System.out.println("Elemen ke-99 : " + data[98]);
174       System.out.println("Elemen ke-100 : " + data[99]);
175       System.out.println("Elemen ke-101 : " + data[100]);
176       System.out.println("Elemen ke-102 : " + data[101]);
177       System.out.println("Elemen ke-103 : " + data[102]);
178       System.out.println("Elemen ke-104 : " + data[103]);
179       System.out.println("Elemen ke-105 : " + data[104]);
180       System.out.println("Elemen ke-106 : " + data[105]);
181       System.out.println("Elemen ke-107 : " + data[106]);
182       System.out.println("Elemen ke-108 : " + data[107]);
183       System.out.println("Elemen ke-109 : " + data[108]);
184       System.out.println("Elemen ke-110 : " + data[109]);
185       System.out.println("Elemen ke-111 : " + data[110]);
186       System.out.println("Elemen ke-112 : " + data[111]);
187       System.out.println("Elemen ke-113 : " + data[112]);
188       System.out.println("Elemen ke-114 : " + data[113]);
189       System.out.println("Elemen ke-115 : " + data[114]);
190       System.out.println("Elemen ke-116 : " + data[115]);
191       System.out.println("Elemen ke-117 : " + data[116]);
192       System.out.println("Elemen ke-118 : " + data[117]);
193       System.out.println("Elemen ke-119 : " + data[118]);
194       System.out.println("Elemen ke-120 : " + data[119]);
195       System.out.println("Elemen ke-121 : " + data[120]);
196       System.out.println("Elemen ke-122 : " + data[121]);
197       System.out.println("Elemen ke-123 : " + data[122]);
198       System.out.println("Elemen ke-124 : " + data[123]);
199       System.out.println("Elemen ke-125 : " + data[124]);
200       System.out.println("Elemen ke-126 : " + data[125]);
201       System.out.println("Elemen ke-127 : " + data[126]);
202       System.out.println("Elemen ke-128 : " + data[127]);
203       System.out.println("Elemen ke-129 : " + data[128]);
204       System.out.println("Elemen ke-130 : " + data[129]);
205       System.out.println("Elemen ke-131 : " + data[130]);
206       System.out.println("Elemen ke-132 : " + data[131]);
207       System.out.println("Elemen ke-133 : " + data[132]);
208       System.out.println("Elemen ke-134 : " + data[133]);
209       System.out.println("Elemen ke-135 : " + data[134]);
210       System.out.println("Elemen ke-136 : " + data[135]);
211       System.out.println("Elemen ke-137 : " + data[136]);
212       System.out.println("Elemen ke-138 : " + data[137]);
213       System.out.println("Elemen ke-139 : " + data[138]);
214       System.out.println("Elemen ke-140 : " + data[139]);
215       System.out.println("Elemen ke-141 : " + data[140]);
216       System.out.println("Elemen ke-142 : " + data[141]);
217       System.out.println("Elemen ke-143 : " + data[142]);
218       System.out.println("Elemen ke-144 : " + data[143]);
219       System.out.println("Elemen ke-145 : " + data[144]);
220       System.out.println("Elemen ke-146 : " + data[145]);
221       System.out.println("Elemen ke-147 : "
```

Kode program StackMain19.java:

```

1  import java.util.Scanner;
2  import java.util.Stack;
3
4  public class StackMain14 {
5      public static void main(String[] args) {
6
7          StackPakaian19 stk19 = new StackPakaian19(5);
8          Scanner sc19 = new Scanner(System.in);
9
10         char pilih;
11
12         do{
13             System.out.print("Jenis: ");
14             String jenis = sc19.nextLine();
15             System.out.print("Warna: ");
16             String warna = sc19.nextLine();
17             System.out.print("Merk: ");
18             String merk = sc19.nextLine();
19             System.out.print("Ukuran: ");
20             String ukuran = sc19.nextLine();
21             System.out.print("Harga: ");
22             double harga = sc19.nextDouble();
23
24             Pakaian19 p19 = new Pakaian19(jenis, warna, merk, ukuran, harga);
25             System.out.print("Apakah anda akan menambahkan data baru ke stack (y/n)? ");
26             pilih = sc19.next().charAt(0);
27             sc19.nextLine();
28             stk19.push(p19);
29         } while (pilih == 'y');
30
31         stk19.getMax();
32         stk19.print();
33         stk19.pop();
34         stk19.peek();
35         stk19.print();
36     }
37 }

```

Hasil run kode program:

```
Data pakaian dengan harga tertinggi: 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

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
```

2. Setiap hari Minggu, Dewi pergi berbelanja ke salah satu supermarket yang berada di area rumahnya. Setiap kali selesai berbelanja, Dewi menyimpan struk belanjanya di dalam laci. Setelah dua bulan, ternyata Dewi sudah mempunyai delapan struk belanja. Dewi berencana mengambil lima struk belanja untuk ditukarkan dengan voucher belanja. Buat sebuah program stack untuk menyimpan data struk belanja Dewi, kemudian lakukan juga proses pengambilan data struk belanja sesuai dengan jumlah struk yang akan ditukarkan dengan voucher. Informasi yang tersimpan pada struk belanja terdiri dari:
- Nomor transaksi
 - Tanggal pembelian
 - Jumlah barang yang dibeli
 - Total harga bayar

Kode program Tugas219.java:

Kode program Tugas2Main19.java:

```
1 import java.text.SimpleDateFormat;
2 import java.util.Date;
3
4 public class Tugas2Main19 {
5
6     public static void main(String[] args) {
7         Date waktu = new Date();
8
9         SimpleDateFormat sdf = new SimpleDateFormat("EEEE,DD MMMM YYYY");
10
11         Tugas219 struk = new Tugas219(8);
12         Tugas219[] belanja = new Tugas219[8];
13         String date = sdf.format(waktu);
14
15         belanja[0] = new Tugas219("MEDSBK-090-445", 6, 2400000, date);
16         belanja[1] = new Tugas219("MEDSBK-080-446", 7, 1500000, date);
17         belanja[2] = new Tugas219("MEDSBK-056-447", 15, 1800000, date);
18         belanja[3] = new Tugas219("MEDSBK-683-448", 6, 190000, date);
19         belanja[4] = new Tugas219("MEDSBK-936-449", 2, 2356000, date);
20         belanja[5] = new Tugas219("MEDSBK-081-441", 5, 800000, date);
21         belanja[6] = new Tugas219("MEDSBK-482-442", 1, 6900000, date);
22         belanja[7] = new Tugas219("MEDSBK-593-443", 7, 4700000, date);
23
24         for (int i = 0; i < struk.size; i++) {
25             struk.push(belanja[i]);
26         }
27         struk.print();
28         System.out.println("\tData yang di-pop |struk 5 minggu|");
29         struk.line(50);
30         for (int i = 0; i < 5; i++) {
31             struk.pop();
32         }
33         struk.print();
34     }
35 }
36
```

Hasil run kode program:

```

                                ISI STACK
=====
Nomor Transaksi : MEDSBK-593-443
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 7
Total          : Rp. 4700000
Nomor Transaksi : MEDSBK-482-442
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 1
Total          : Rp. 6900000
Nomor Transaksi : MEDSBK-081-441
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 5
Total          : Rp. 800000
Nomor Transaksi : MEDSBK-936-449
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 2
Total          : Rp. 2356000
Nomor Transaksi : MEDSBK-683-448
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 6
Total          : Rp. 190000
Nomor Transaksi : MEDSBK-056-447
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 15
Total          : Rp. 1800000
Nomor Transaksi : MEDSBK-080-446
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 7
Total          : Rp. 1500000
Nomor Transaksi : MEDSBK-090-445
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 6
Total          : Rp. 2400000
```


Data yang di-pop |struk 5 minggu|

```
=====
Nomor Transaksi : MEDSBK-593-443
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 7
Total          : Rp. 4700000
Nomor Transaksi : MEDSBK-482-442
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 1
Total          : Rp. 6900000
Nomor Transaksi : MEDSBK-081-441
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 5
Total          : Rp. 800000
Nomor Transaksi : MEDSBK-936-449
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 2
Total          : Rp. 2356000
Nomor Transaksi : MEDSBK-683-448
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 6
Total          : Rp. 190000
```

ISI STACK

```
=====
Nomor Transaksi : MEDSBK-056-447
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 15
Total          : Rp. 1800000
Nomor Transaksi : MEDSBK-080-446
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 7
Total          : Rp. 1500000
Nomor Transaksi : MEDSBK-090-445
Tanggal        : Friday,96 April 2024
Jumlah Barang  : 6
Total          : Rp. 2400000
```