

Laporan Praktikum Alogaritma dan Struktur Data

Jobsheet 9 : Stack

Alogaritma dan Struktur Data

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



Eka Putri Natalya Kabelen

2341760107

SIB 1E

PROGRAM STUDI SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

7.2. Praktikum 1

7.2.1 Langkah-langkah Percobaan

1. Buat class Stack.java

```
J Stack11.java > Stack11
1  /**
2  * Stack11
3  */
4  public class Stack11 {
```

2. Tulis kode untuk membuat atribut dan konstruktor pada class Stack sebagai berikut

```
6      int data[];
7      int size;
8      int top;
9
10     public Stack11(int size) {
11         this.size = size;
12         data = new int[size];
13         top = -1;
14     }
```

3. Lalu tambahkan method isFull() dan isEmpty() pada class Stack sebagai berikut

```
15     public boolean isFul() {
16         if (top == size-1) {
17             return true;
18         } else {
19             return false;
20         }
21     }
22
23     public boolean isEmpty() {
24         if (top == -1) {
25             return true;
26         } else {
27             return false;
28         }
29     }
```

4. Tambahkan method `push(int data)` dan `pop()` sebagai berikut

```
31     public void push(int dt) {
32         if (!isFull()) {
33             top++;
34             data[top] = dt;
35         } else {
36             System.out.println(x:"Stack penuh");
37         }
38     }
39
40     public void pop() {
41         if (!isEmpty()) {
42             int x = data[top];
43             top--;
44             System.out.println("Data yang dikeluarkan dari stack: "+x);
45         } else {
46             System.out.println(x:"Stack masih kosong");
47         }
48     }
```

5. Tambahkan method `peek()` sebagai berikut

```
50     public void peek() {
51         System.out.println("Elemen teratas stack: "+data[top]);
52     }
```

6. Tambahkan method `print()` dan `clear()` sebagai berikut

```
54     public void print() {
55         System.out.println(x:"Isi stack: ");
56         for(int i = top; i >= 0; i--) {
57             System.out.println(data[i]+" ");
58         }
59         System.out.println(x:" ");
60     }
61     public void clear() {
62         if (!isEmpty()) {
63             for (int i = top; i >= 0; i--) {
64                 top--;
65             }
66             System.out.println(x:"Stack sudah dikosongkan");
67         } else {
68             System.out.println(x:"Stack masih kosong");
69         }
70     }
71 }
```

7. Buat file StackDemo.java untuk mengimplementasikan class StackDemo yang berisi fungsi main untuk membuat objek Stack dan mengoperasikan method-method pada class Stack.

```
J StackDemo.java > StackDemo > main(String[])
1  public class StackDemo {
2
3      Run | Debug
4      public static void main(String[] args) {
5          Stack11 stack11 = new Stack11(size:10);
6          stack11.push(dt:8);
7          stack11.push(dt:12);
8          stack11.push(dt:18);
9          stack11.print();
10         stack11.pop();
11         stack11.peek();
12         stack11.pop();
13         stack11.push(-5);
14         stack11.print();
15     }
```

8. Compile dan run class StackDemo.

```
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 :

Pemanggilan method `isEmpty()` berguna untuk memeriksa apakah struktur data tersebut sudah kosong sebelum melakukan operasi `pop()`. Jika tidak ada pemanggilan `isEmpty()` sebelum melakukan `pop()`, maka ada risiko bahwa akan terjadi error atau pengecualian jika mencoba melakukan operasi `pop()` pada struktur data yang sudah kosong. Hal ini bisa menyebabkan program berhenti berjalan atau menghasilkan hasil yang tidak diinginkan.

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

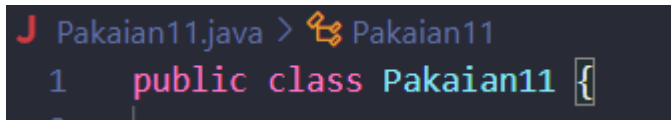
Jawab :

Perbedaan antara kedua method ini adalah bahwa method `peek()` hanya melihat nilai dari elemen teratas tanpa menghapusnya, sedangkan pada method `pop()` dapat menghapus dan mengembalikan nilai dari elemen teratas tumpukan.

7.3. Praktikum 2

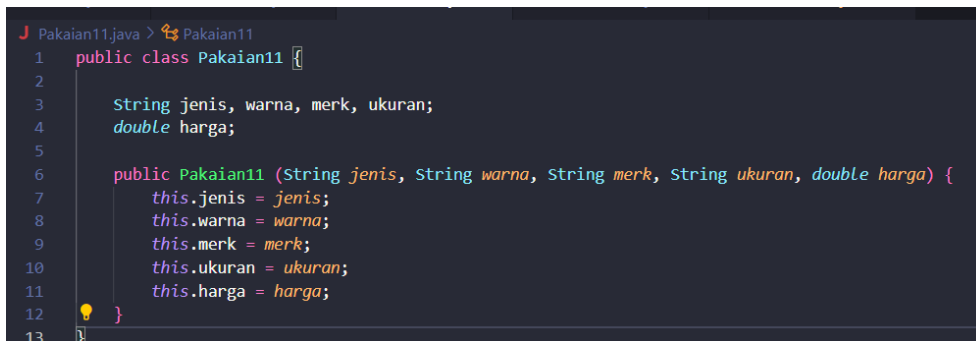
7.3.1. Langkah-langkah Percobaan

1. Buat class baru dengan nama Pakaian.




```
J Pakaian11.java > Pakaian11
1 public class Pakaian11 {
```

2. Tambahkan atribut-atribut Pakaian seperti pada Class Diagram Pakaian, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.



```
J Pakaian11.java > Pakaian11
1 public class Pakaian11 {
2
3     String jenis, warna, merk, ukuran;
4     double harga;
5
6     public Pakaian11 (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 }
```

3. Buat class baru dengan nama Stack. Kemudian tambahkan atribut dan konstruktor seperti gambar berikut ini.



```
J StackPakaian.java > StackPakaian
1 public class StackPakaian {
2
3     int size;
4     int top;
5     Pakaian11 data[];
6
7     public StackPakaian (int size) {
8         this.size = size;
9         data = new Pakaian11[size];
10        top = -1;
11    }
```

4. Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah stack kosong

```

13      public boolean isEmpty() {
14          if (top == -1) {
15              return true;
16          } else {
17              return false;
18          }
19      }
20

```

5. Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah stack sudah terisi penuh.

```

21      public boolean isFull() {
22          if (top == size -1) {
23              return true;
24          } else {
25              return false;
26          }
27      }

```

6. Buat method push bertipe void untuk menambahkan isi elemen stack dengan parameter pkn yang berupa object Pakaian

```

29      public void push(Pakaian11 pkn11) {
30          if (!isFull()) {
31              top++;
32              data[top] = pkn11;
33          } else {
34              System.out.println(x:"Isi Stack penuh!!");
35          }
36      }
37

```

7. Buat method Pop bertipe void untuk mengeluarkan isi elemen stack. Karena satu elemen stack terdiri dari beberapa informasi (jenis, warna, merk, ukuran, dan harga), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut

```

38      public void pop(){
39          if (!isEmpty()){
40              Pakaian11 x = data[top];
41              top--;
42              System.out.println("Data yang keluar: " + x.jenis + " " + x.warna + " " + x.merk + " " + x.ukuran + " " + x.harga);
43          } else {
44              System.out.println(x:"Stack masih kosong");
45          }
46      }

```

8. Buat method peek bertipe void untuk memeriksa elemen stack pada posisi paling atas

```

public void peek() {
    System.out.println("Elemen teratas: " + data[top].jenis + " " + data[top].warna + " " + data[top].merk + " " +
        data[top].ukuran + " " + data[top].harga);
}

```

9. Buat method print bertipe void untuk menampilkan seluruh elemen pada stack

```

public void print() {
    System.out.println(x:"Isi stack: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i].jenis + " " + data[i].warna + " " + data[i].merk + " " + data[i].ukuran + " " + data[i].harga + " ");
    }
    System.out.println(x:"");
}

```

10. Buat method clear bertipe void untuk menghapus seluruh isi stack.

```

61     public void clear() {
62         if (!IsEmpty()) {
63             for (int i = top; i >= 0; i--) {
64                 top--;
65             }
66             System.out.println(x:"Stack sudah dikosongkan");
67         } else {
68             System.out.println(x:"Stack masih kosong");
69         }
70     }
71 }

```

11. Selanjutnya, buat class baru dengan nama StackMain. Buat fungsi main, kemudian lakukan instansiasi objek dari class Stack dengan nama stk dan nilai parameternya adalah 5 dan deklarasikan Scanner dengan nama sc .

```

J StackMain.java > StackMain
1  import java.util.Scanner;
2  import java.util.Stack;
3
4  public class StackMain {
5
6      Run | Debug
7      public static void main(String[] args) {
8
9          StackPakaian stk = new StackPakaian(size:5);
10         Scanner sc = new Scanner(System.in);

```

12. Tambahkan kode berikut ini untuk menerima input data Pakaian, kemudian semua informasi tersebut dimasukkan ke dalam stack


```

11     char pilih;
12
13     do{
14         System.out.print(s:"Jenis: ");
15         String jenis = sc.nextLine();
16         System.out.print(s:"Warna: ");
17         String warna = sc.nextLine();
18         System.out.print(s:"Merk: ");
19         String merk = sc.nextLine();
20         System.out.print(s:"Ukuran: ");
21         String ukuran = sc.nextLine();
22         System.out.print(s:"Harga: ");
23         double harga = sc.nextDouble();
24
25         Pakaian11 p11 = new Pakaian11 (jenis, warna, merk, ukuran, harga);
26         System.out.print(s:"Apakah anda akan menambahkan data baru ke stack (y/n)? ");
27         pilih = sc.next().charAt(index:0);
28         sc.nextLine();
29         stk.push(p11);
30     } while (pilih == 'y');

```

13. Lakukan pemanggilan method print, method pop, dan method peek dengan urutan sebagai berikut.

```

32     stk.print();
33     stk.pop();
34     stk.peek();
35     stk.print();
36 }
37

```

14. Compile dan jalankan class StackMain, kemudian amati hasilnya

```

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
Isi stack:
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0

Data yang keluar: Kemeja Putih Styves XL 127000.0
Elemen teratas: Kaos Hitam Nevada M 85000.0
Isi stack:
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 :

Banyak data pakaian yang dapat ditampung didalam stack berjumlah 5 elemen, adapun potongan kodenya terdapat pada Main class "stackMain" berikut ini

```
7 StackPakaian stk = new StackPakaian(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 :

Data yang tersimpan pada variabel p adalah data data yang dimasukkan melalui scanner

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

Jawab :

Berfungsi untuk melakukan perulangan agar nantinya user dapat menginputkan data yang meliputi judul, nama pengarang, tahun terbit, jumlah halaman, dan harga. Maka setelah melakukan penginputan, user akan diberi pilihan untuk menambahkan data baru atau tidak, jika "y" maka user akan diminta untuk menginputkan data data lagi, dan jika user memilih "n" maka program akan berhenti.

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

Jawab :

Hasil run dari modifikasi kode program tersebut

```
Pilih operasi stack yang di inginkan:
1. Fungsi Pop
2. Fungsi Peek
3. Print
4. Keluar
Masukkan pilihan anda: 1
Data yang keluar: Kemeja Putih Styves XL 185000.0
```

```
Pilih operasi stack yang di inginkan:
1. Fungsi Pop
2. Fungsi Peek
3. Print
4. Keluar
Masukkan pilihan anda: 2
Elemen teratas: 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: 3
Isi stack:
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.
```

Code program dari hasil modifikasi program

```
1  import java.util.Scanner;
2  import java.util.Stack;
3
4  public class StackMain11 {
5
6      public static void main(String[] args) {
7
8          StackPakaian stk = new StackPakaian(5);
9          Scanner sc11 = new Scanner(System.in);
10
11          char pilih;
12
13          do{
14              System.out.print("Jenis: ");
15              String jenis = sc11.nextLine();
16              System.out.print("Warna: ");
17              String warna = sc11.nextLine();
18              System.out.print("Merk: ");
19              String merk = sc11.nextLine();
20              System.out.print("Ukuran: ");
21              String ukuran = sc11.nextLine();
22              System.out.print("Harga: ");
23              double harga = sc11.nextDouble();
24
25              Pakaian11 p11 = new Pakaian11(jenis, warna, merk, ukuran, harga);
26              System.out.print("Apakah anda akan menambahkan data baru ke stack (y/n)? ");
27              pilih = sc11.next().charAt(0);
28              sc11.nextLine();
29              stk.push(p11);
30          } while (pilih == 'y');
31
32          int input;
33          boolean lanjut = true;
34
35          while (lanjut) {
36              System.out.println("\nPilih operasi stack yang di inginkan: ");
37              System.out.println("1. Fungsi Pop");
38              System.out.println("2. Fungsi Peek");
39              System.out.println("3. Print");
40              System.out.println("4. Keluar");
41              System.out.print("Masukkan pilihan anda: ");
42              input = sc11.nextInt();
43
44              switch (input) {
45                  case 1:
46                      stk.pop();
47                      break;
48                  case 2:
49                      stk.peek();
50                      break;
51                  case 3:
52                      stk.print();
53                      break;
54                  case 4:
55                      lanjut = false;
56                      break;
57                  default:
58                      System.out.println("Pilihan anda salah harap ulangi lagi");
59              }
60          }
61
62          System.out.println("Program selesai.");
63          sc11.close();
64      }
65  }
```

7.4. Praktikum 3

7.4.1. Langkah-langkah Percobaan

1. Buat class baru dengan nama Postfix. Tambahkan atribut n, top, dan stack sesuai diagram class Postfix tersebut.

```
J Postfix11.java > Postfix11 > Postfix11(int)
1 public class Postfix11 {
```

2. Tambahkan pula konstruktor berparameter seperti gambar berikut ini.

```
3         int n, top;
4         char[] stack;
5
6         public Postfix11 (int total) {
7             n = total;
8             top = -1;
9             stack = new char[n];
10            push (c: '(');
11        }
```

3. Buat method push dan pop bertipe void

```
13        private void push(char c) {
14            top++;
15            stack[top] = c;
16        }
17
18        public char pop() {
19            char item = stack[top];
20            top--;
21            return item;
22        }
```

4. Buat method IsOperand dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operand.

```
public boolean IsOperand (char c) {
    if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || c == ' ' || c == '.') {
        return true;
    } else {
        return false;
    }
}
```

5. Buat method IsOperator dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operator.

```

public boolean IsOperand (char c) {
    if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || c == ' ' || c == '.') {
        return true;
    } else {
        return false;
    }
}

public boolean IsOperator(char c) {
    if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {
        return true;
    } else {
        return false;
    }
}

```

6. Buat method derajat yang mempunyai nilai kembalian integer untuk menentukan derajat operator.

```

public int derajat (char c) {
    switch (c) {
        case '^':
            return 3;
        case '&':
            return 2;
        case '/':
            return 2;
        case '*':
            return 2;
        case '-':
            return 1;
        case '+':
            return 1;
        default:
            return 0;
    }
}

```

7. Buat method konversi untuk melakukan konversi notasi infix menjadi notasi postfix dengan cara mengecek satu persatu elemen data pada String Q sebagai parameter masukan.

```

59     public String konversi(String Q) {
60         String P = "";
61         char c;
62         for (int i = 0; i < n; i++) {
63             c = Q.charAt(i);
64             if (IsOperand(c)) {
65                 P = P + c;
66             }
67             if (c == '(') {
68                 push(c);
69             }
70             if (c == ')') {
71                 while (stack[top] != '(') {
72                     P = P + pop();
73                 }
74                 pop();
75             }
76             if (IsOperator(c)) {
77                 while (top >= 0 && derajat(stack[top]) >= derajat(c)) {
78                     P = P + pop();
79                 }
80                 push(c);
81             }
82         }
83         return P;
84     }
85 }

```

8. Selanjutnya, buat class baru dengan nama PostfixMain. Buat class main, kemudian buat variabel P dan Q. Variabel P digunakan untuk menyimpan hasil akhir notasi postfix setelah dikonversi, sedangkan variabel Q digunakan untuk menyimpan masukan dari pengguna berupa ekspresi matematika dengan notasi infix. Deklarasikan variabel Scanner dengan nama sc, kemudian panggil fungsi built-in trim yang digunakan untuk menghapus adanya spasi di depan atau di belakang teks dari teks persamaan yang dimasukkan oleh pengguna.

```

J PostfixMain11.java > PostfixMain11
1  import java.util.Scanner;
2
3  public class PostfixMain11 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner sc11 = new Scanner(System.in);
8          String P, Q;
9          System.out.println("Masukkan ekspresi matematika (infix): ");
10         Q = sc11.nextLine();
11         Q = Q.trim();
12         Q = Q + ")";

```

9. Buat variabel total untuk menghitung banyaknya karakter pada variabel Q.

```

13         int total = Q.length();

```

10. Lakukan instansiasi objek dengan nama post dan nilai parameternya adalah total. Kemudian panggil method konversi untuk melakukan konversi notasi infix Q menjadi notasi postfix P.

```

15     Postfix11 post = new Postfix11(total);
16     P = post.konversi(Q);
17     System.out.println("Posttix: "+P);
18 }
19

```

11. Compile dan jalankan class PostfixMain dan amati hasilnya

```

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

```

7.4.3. Pertanyaan

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

Jawab :

Pada class "postfix", method derajat memiliki alur seperti berikut ini ketika operator tersebut adalah '^' maka akan mereturn nilai 3, jika operator '%','/','*' maka akan mereturn nilai 2. dan sedangkan operator '-', '+' maka akan mereturn nilai 1 dengan menggunakan perintah switch case yang telah tersedia.

2. Apa fungsi kode program berikut?

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

Jawab :

Fungsi kode program tersebut yaitu Variabel c digunakan untuk menyimpan data char i ke dalam variabel Q dengan menggunakan perintah charAt(i).

3. Jalankan kembali program tersebut, masukkan ekspresi $5*4^{(1+2)}\%$ 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 Postfix11.konversi(Postfix11.java:71)
    at PostfixMain11.main(PostfixMain11.java:16)
```

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

Jawab :

Karena ketika terdapat tanda kurung pada ekspresi matematika, maka tanda kurung tersebut akan langsung di pop tanpa dimasukkan kedalam postfix.

7.5. Tugas

1. Perhatikan dan gunakan kembali kode program pada Praktikum2. 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

Berikut merupakan method getMax pada class StackPakaian

```
public void getMax() {
    if (!isEmpty()) {
        Pakaian11 maxPakaian = data[0];
        for (int i = 1; i <= top; i++) {
            if (data[i].harga > maxPakaian.harga) {
                maxPakaian = data[i];
            }
        }
        System.out.println("Data pakaian dengan harga tertinggi:" + maxPakaian.jenis + " " + maxPakaian.warna + " " + maxPakaian.merk + " " +
            maxPakaian.ukuran + " " + maxPakaian.harga);
    } else {
        System.out.println(x:"Stack masih kosong");
    }
}
```

Berikut merupakan pemanggilan method getMax pada StackMain

```
Pakaian11 p11 = new Pakaian11(jenis, warna, merk, ukuran, harga);
System.out.print(s:"Apakah anda akan menambahkan data baru ke stack (y/n)? ");
pilih = sc.next().charAt(index:0);
sc.nextLine();
stk.push(p11);
} while (pilih == 'y');

stk.getMax();
stk.print();
stk.pop();
stk.peek();
stk.print();
}
```

Hasil Run dari code program

```
Apakah anda akan menambahkan data baru ke stack (y/n)? n
Data pakaian dengan harga tertinggi:jaket putih the north face L 1750000.0
Isi stack:
kaos biru jj jeans L 150000.0
jaket putih the north face L 1750000.0
kemeja merah nevada L 460000.0

Data yang keluar: kaos biru jj jeans L 150000.0
Elemen teratas: jaket putih the north face L 1750000.0
Isi stack:
jaket putih the north face L 1750000.0
kemeja merah nevada L 460000.0
```

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

Tampilkan informasi struk belanja yang masih tersimpan di dalam stack!

JAWAB

CODE PROGRAM


```
1 import java.text.SimpleDateFormat;
2 import java.util.Date;
3
4 public class Tugas2Main {
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         Tugas2 struk = new Tugas2(8);
12         Tugas2[] belanja = new Tugas2[8];
13         String date = sdf.format(waktu);
14
15         belanja[0] = new Tugas2("MEDSBK-090-445", 6, 2400000, date);
16         belanja[1] = new Tugas2("MEDSBK-080-446", 7, 1500000, date);
17         belanja[2] = new Tugas2("MEDSBK-056-447", 15, 1800000, date);
18         belanja[3] = new Tugas2("MEDSBK-683-448", 6, 190000, date);
19         belanja[4] = new Tugas2("MEDSBK-936-449", 2, 2356000, date);
20         belanja[5] = new Tugas2("MEDSBK-081-441", 5, 800000, date);
21         belanja[6] = new Tugas2("MEDSBK-482-442", 1, 6900000, date);
22         belanja[7] = new Tugas2("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 }
```

Hasil run dari code program diatas

```
ISI STACK
=====
Nomor Transaksi : MEDSBK-593-443
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 7
Total          : Rp. 4700000
Nomor Transaksi : MEDSBK-482-442
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 1
Total          : Rp. 6900000
Nomor Transaksi : MEDSBK-081-441
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 5
Total          : Rp. 800000
Nomor Transaksi : MEDSBK-936-449
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 2
Total          : Rp. 2356000
Nomor Transaksi : MEDSBK-683-448
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 6
Total          : Rp. 190000
Nomor Transaksi : MEDSBK-056-447
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 15
Total          : Rp. 1800000
Nomor Transaksi : MEDSBK-080-446
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 7
Total          : Rp. 1500000
Nomor Transaksi : MEDSBK-090-445
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 6
Total          : Rp. 2400000

Data yang di-pop |struk 5 minggu|
=====
```

```
=====
Nomor Transaksi : MEDSBK-593-443
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 7
Total          : Rp. 4700000
Nomor Transaksi : MEDSBK-482-442
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 1
Total          : Rp. 6900000
Nomor Transaksi : MEDSBK-081-441
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 5
Total          : Rp. 800000
Nomor Transaksi : MEDSBK-936-449
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 2
Total          : Rp. 2356000
Nomor Transaksi : MEDSBK-683-448
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 6
Total          : Rp. 190000

ISI STACK
=====
Nomor Transaksi : MEDSBK-056-447
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 15
Total          : Rp. 1800000
Nomor Transaksi : MEDSBK-080-446
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 7
Total          : Rp. 1500000
Nomor Transaksi : MEDSBK-090-445
Tanggal        : Thursday,95 April 2024
Jumlah Barang  : 6
Total          : Rp. 2400000
```