

Laporan Teori Algoritma dan Struktur Data

Jobsheet 9 – Stack

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



Nama : Annisa
Nim : 2341760032
Prodi : D-IV Sistem Informasi Bisnis
Absen : 04

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

9.2. Praktikum 1

9.2.1 Langkah-langkah Percobaan

1. Buat folder dengan nama Praktikum07. Buat file Stack.java.
2. Tulis kode untuk membuat atribut dan konstruktor pada class Stack sebagai berikut:

```
1  /**
2   * Stack04
3   */
4
5  public class Stack04 {
6
7      int data[];
8      int size;
9      int top;
10
11     public Stack04(int size) {
12         this.size = size;
13         data = new int[size];
14         top = -1;
15     }
```

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

```
1  public boolean isFul() {
2      if (top == size-1) {
3          return true;
4      } else {
5          return false;
6      }
7  }
8
9  public boolean isEmpty() {
10     if (top == -1) {
11         return true;
12     } else {
13         return false;
14     }
15 }
```

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

```
1  public void push(int dt) {
2      if (!isFul()) {
3          top++;
4          data[top] = dt;
5      } else {
6          System.out.println("Stack penuh");
7      }
8  }
9
10 public void pop() {
11     if (!isEmpty()) {
12         int x = data[top];
13         top--;
14         System.out.println("Data yang dikeluarkan dari stack: "+x);
15     } else {
16         System.out.println("Stack masih kosong");
17     }
18 }
19
```

5. Tambahkan method peek() sebagai berikut:

```
1 public void peek() {
2     System.out.println("Elemen teratas stack: "+data[top]);
3 }
4
```

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

```
1 public void print() {
2     System.out.println("Isi stack: ");
3     for(int i = top; i >= 0; i--) {
4         System.out.println(data[i]+" ");
5     }
6     System.out.println(" ");
7 }
8 public void clear() {
9     if (!isEmpty()) {
10         for (int i = top; i >= 0; i--) {
11             top--;
12         }
13         System.out.println("Stack sudah dikosongkan");
14     } else {
15         System.out.println("Stack masih kosong");
16     }
17 }
18 }
```

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

```
1 /**
2  * StackDemo04
3  */
4 public class StackDemo04 {
5
6     public static void main(String[] args) {
7         Stack04 stack04 = new Stack04(10);
8         stack04.push(8);
9         stack04.push(12);
10        stack04.push(18);
11        stack04.print();
12        stack04.pop();
13        stack04.peek();
14        stack04.pop();
15        stack04.push(-5);
16        stack04.print();
17    }
18 }
```

8. Compile dan run class StackDemo.

9.2.2 Verifikasi Hasil Percobaan

```
6b2de1205\redhat.java\jdt_ws\ASD_9_6fcc8fef\bin" StackDemo04 "  
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  
  
C:\Users\user\Documents\ASD_9>
```

9.2.3 Pertanyaan

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

karena untuk menghindari pengecualian. Method `isEmpty()` digunakan untuk mengecek apakah stack masih kosong atau tidak. Apabila tidak ada pemanggilan method tersebut, maka struktur data mungkin akan mencoba melakukan penghapusan elemen dari struktur data yang kosong. Ini bisa menyebabkan kesalahan atau pengecualian, seperti `"IndexError"` atau `"NoSuchElementException"`.

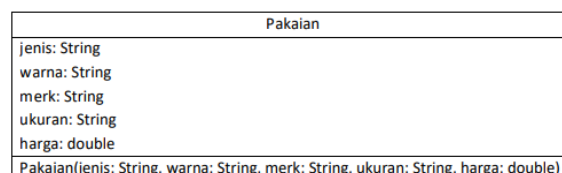
2. Jelaskan perbedaan antara method `peek()` dengan method `pop()` pada class `Stack`.

metode `peek()` digunakan untuk melihat elemen teratas dari stack tanpa menghapusnya. Saat memanggil `peek()`, elemen teratas dari stack hanya dilihat dan tidak dihapus dari stack. Metode `pop()` digunakan untuk menghapus dan mengembalikan elemen teratas dari stack. Saat Anda memanggil `pop()`, elemen teratas stack dihapus dari stack, dan nilai tersebut dikembalikan atau digunakan dalam operasi selanjutnya.

9.3. Praktikum 2

9.3.1. Langkah-langkah Percobaan

1. Perhatikan Diagram Class Pakaian berikut ini:



Berdasarkan diagram class tersebut, akan dibuat program class Pakaian dalam Java.

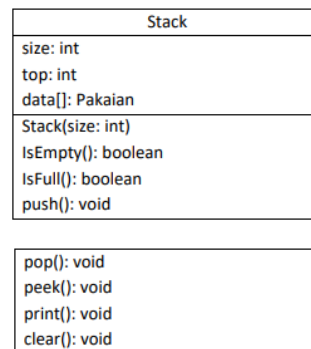
2. Buat class baru dengan nama Pakaian.

```
1 public class Pakaian04 {
```

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

```
1 String jenis, warna, merk, ukuran;
2 double harga;
3
4 public Pakaian04 (String jenis, String warna, String merk, String ukuran, double harga) {
5     this.jenis = jenis;
6     this.warna = warna;
7     this.merk = merk;
8     this.ukuran = ukuran;
9     this.harga = harga;
10 }
11 }
```

4. Setelah membuat class Pakaian, selanjutnya perlu dibuat class Stack yang berisi atribut dan method sesuai diagram Class Stack berikut ini:



Keterangan: Tipe data pada variabel data menyesuaikan dengan data yang akan disimpan di dalam Stack. Pada praktikum ini, data yang akan disimpan merupakan array of object dari Pakaian, sehingga tipe data yang digunakan adalah Pakaian

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

```
1 /**
2  * StackPakaian04
3  */
4 public class StackPakaian04 {
5
6     int size;
7     int top;
8     Pakaian04 data[];
9
10    public StackPakaian04 (int size) {
11        this.size = size;
12        data = new Pakaian04[size];
13        top = -1;
14    }
```

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

```
1 public boolean IsEmpty() {
2     if (top == -1) {
3         return true;
4     } else {
5         return false;
6     }
7 }
```

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

```
1 public boolean isFull() {
2     if (top == size -1) {
3         return true;
4     } else {
5         return false;
6     }
7 }
```

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

```
1 public void push(Pakaian04 pkn04) {
2     if (!isFull()) {
3         top++;
4         data[top] = pkn04;
5     } else {
6         System.out.println("Isi Stack penuh!!");
7     }
8 }
9 }
```

9. 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

```
1 public void pop(){
2     if (!isEmpty()){
3         Pakaian04 x = data[top];
4         top--;
5         System.out.println("Data yang keluar: " + x.jenis + " " + x.warna + " " + x.merk + " " + x.ukuran + " " + x.harga);
6     } else {
7         System.out.println("Stack masih kosong");
8     }
9 }
```

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

```
1 public void peek() {
2     System.out.println("Elemen teratas: " + data[top].jenis + " " + data[top].warna + " " + data[top].merk + " " +
3     data[top].ukuran + " " + data[top].harga);
4 }
```

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

```
1 public void print() {
2     System.out.println("Isi stack: ");
3     for (int i = top; i >= 0; i--) {
4         System.out.println(data[i].jenis + " " + data[i].warna + " " + data[i].merk + " " + data[i].ukuran + " " + data[i].harga + " ");
5     }
6     System.out.println("");
7 }
8 }
```

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

```
1 public void clear() {
2     if (!isEmpty()) {
3         for (int i = top; i >= 0; i--) {
4             top--;
5         }
6         System.out.println("Stack sudah dikosongkan");
7     } else {
8         System.out.println("Stack masih kosong");
9     }
10 }
11 }
```

13. 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.

```
1 public static void main(String[] args) {
2     StackPakaian04 stk = new StackPakaian04(5);
3 }
```

14. Deklarasikan Scanner dengan nama sc

```
1 Scanner sc04 = new Scanner(System.in);
```

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

```
1
2 char pilih;
3 do {
4     System.out.print("Jenis: ");
5     String jenis = sc04.nextLine();
6     System.out.print("Warna: ");
7     String warna = sc04.nextLine();
8     System.out.print("Merk: ");
9     String merk = sc04.nextLine();
10    System.out.print("Ukuran: ");
11    String ukuran = sc04.nextLine();
12    System.out.print("Harga: ");
13    double harga = sc04.nextDouble();
14
15    Pakaian04 p = new Pakaian04(jenis, warna, merk, ukuran, harga);
16    System.out.print("Apakah Anda akan menambahkan data baru ke stack (y/n?)");
17    pilih = sc04.next().charAt(0);
18    sc04.nextLine(); //digunakan untuk mengembalikan karakter new line
19    stk.push(p);
20 } while (pilih == 'y');
```

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

```
1 stk.print();
2 stk.pop();
3 stk.peek();
4 stk.print();
```

17. Compile dan jalankan class StackMain, kemudian amati hasilnya.

9.3.2. Verifikasi Hasil Percobaan

```
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

C:\Users\user\Documents\ASD 9>
```


9.3.3. Pertanyaan

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

Banyak data pakaian yang dapat ditampung di dalam stack berjumlah 5



```
1 public static void main(String[] args) {  
2     StackPakaian04 stk = new StackPakaian04(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);`

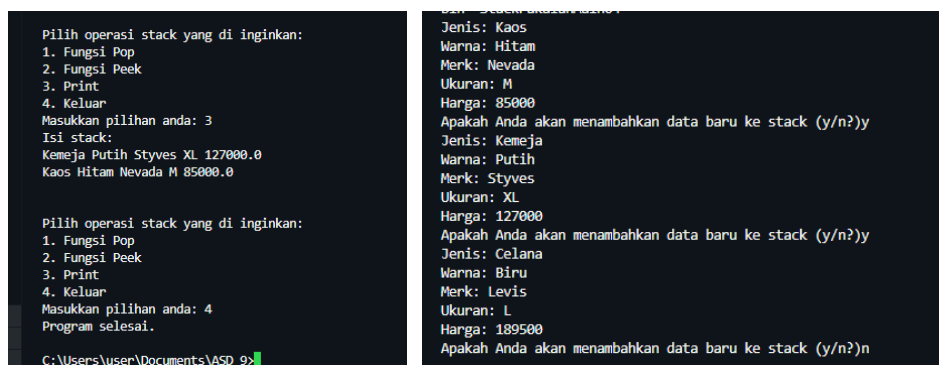
Data yang tersimpan pada variable p adalah data-data yang dimasukkan melalui Scanner

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

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 operasi operasi pada stack (push, pop, peek, atau print) melalui pilihan menu program dengan memanfaatkan kondisi IF-ELSE atau SWITCH-CASE!

Berikut merupakan hasil modifikasi kode program



```
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.  
C:\Users\User\Documents\ASD 0>  
  
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
```

9.4. Praktikum 3

9.4.1. Percobaan

1. Perhatikan Diagram Class berikut ini:

Postfix
n: int
top: int
stack: char[]
Postfix(total: int)
push(c: char): void
pop(): void
IsOperand(c: char): boolean
IsOperator(c: char): boolean
derajat(c: char): int
konversi(Q: String): string

Berdasarkan diagram class tersebut, akan dibuat program class Postfix dalam Java

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



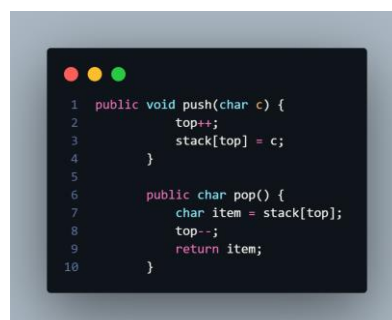
```
1 public class Postfix04 {
2     int n, top;
3     char[] stack;
```

3. Tambahkan pula konstruktor berparameter seperti gambar berikut ini.



```
1 public Postfix04 (int total) {
2     n = total;
3     top = -1;
4     stack = new char[n];
5     push('(');
6 }
```

4. Buat method push dan pop bertipe void.



```
1 public void push(char c) {
2     top++;
3     stack[top] = c;
4 }
5
6 public char pop() {
7     char item = stack[top];
8     top--;
9     return item;
10 }
```

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

```
1 public boolean IsOperand (char c) {
2     if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || c == ' ' || c == '.') {
3         return true;
4     } else {
5         return false;
6     }
7 }
```

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

```
1 public boolean IsOperator(char c) {
2     if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {
3         return true;
4     } else {
5         return false;
6     }
7 }
```

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

```
1 public int derajat (char c) {
2     switch (c) {
3         case '^':
4             return 3;
5         case '&':
6             return 2;
7         case '/':
8             return 2;
9         case '*':
10            return 2;
11        case '-':
12            return 1;
13        case '+':
14            return 1;
15        default:
16            return 0;
17    }
18 }
```

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

```

1  public String konversi(String Q) {
2      String P = "";
3      char c;
4      for (int i = 0; i < n; i++) {
5          c = Q.charAt(i);
6          if (IsOperand(c)) {
7              P = P + c;
8          }
9          if (c == '(') {
10             push(c);
11          }
12          if (c == ')') {
13              while (stack[top] != '(') {
14                  P = P + pop();
15              }
16              pop();
17          }
18          if (IsOperator(c)) {
19              while (top >= 0 && derajat(stack[top]) >= derajat(c)) {
20                  P = P + pop();
21              }
22              push(c);
23          }
24      }
25      return P;
26  }
27
28  }
29
30

```

9. 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.

```

1  public class PostfixMain04 {
2
3      public static void main(String[] args) {
4          Scanner sc04 = new Scanner(System.in);
5          String P, Q;
6          System.out.println("Masukkan ekspresi matematika (infix): ");
7          Q = sc04.nextLine();
8          Q = Q.trim();
9          Q = Q + " ";

```

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

```

1  int total = Q.length();

```

11. 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.

```
1 Postfix04 post = new Postfix04(total);
2 P = post.konversi(Q);
3 System.out.println("Postfix:"+P);
4 }
5 }
```

12. Compile dan jalankan class PostfixMain dan amati hasilnya.

9.4.2. Verifikasi Hasil Percobaan

```
D:\PostfixMain04
Masukkan ekspresi matematika (infix):
a+b*(c+d-e)/f
Postfix:abcd+e-*f/+
C:\Users\user\Documents\ASD 9>
```

9.4.3. Pertanyaan

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

Pada class "postix", 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);
```

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 $5*4^{(1+2)}\%3$. Tampilkan hasilnya! 4. Pada soal nomor 3, mengapa tanda kurung tidak ditampilkan pada hasil konversi? Jelaskan!

```
C:\Users\user\Documents\ASD 9>
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 Postfix04.konversi(Postfix04.java:73)
    at PostfixMain04.main(PostfixMain04.java:16)
C:\Users\user\Documents\ASD 9>
```

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

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

9.5. 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!

Berikut merupakan kode program yang telah dimodifikasi dari praktikum 2

Gambar di bawah merupakan getMax yang terdapat pada class StackPakaian04

```
1 public void getMax() {
2     if (!isEmpty()) {
3         Pakaian04 maxPakaian = data[0];
4         for (int i = 1; i <= top; i++) {
5             if (data[i].harga > maxPakaian.harga) {
6                 maxPakaian = data[i];
7             }
8         }
9         System.out.println("Data pakaian dengan harga tertinggi:" + maxPakaian.jenis + " " + maxPakaian.warna + " " + maxPakaian.merk + " " +
10             maxPakaian.ukuran + " " + maxPakaian.harga);
11     } else {
12         System.out.println("Stack masih kosong");
13     }
14 }
15
16 }
```

Gambar di bawah merupakan pemanggilan method getMax pada class StackPakaianMain04

```
1 Pakaian04 p04 = new Pakaian04(jenis, warna, merk, ukuran, harga);
2 System.out.print("Apakah anda akan menambahkan data baru ke stack (y/n)? ");
3 pilih = sc14.next().charAt(0);
4 sc14.nextLine();
5 stk14.push(p04);
6 } while (pilih == 'y');
7
8 stk14.getMax();
9 stk14.print();
10 stk14.pop();
11 stk14.peek();
12 stk14.print();
13
14 }
```

Gambar di bawah ini merupakan hasil running dari praktikum 2 yang telah dimodifikasi

```
bin\StackPakaianMain04
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)? t
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
Kaos Hitam Nevada M 85000.0
```

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**

Berikut merupakan kode program dari tugas 2

[illegible]



```
1  import java.text.SimpleDateFormat;
2  import java.util.Date;
3
4  /**
5   * Tugas2Main_04
6   */
7  public class Tugas2Main_04 {
8      public static void main(String[] args) {
9          Date waktu = new Date();
10
11          SimpleDateFormat sdf = new SimpleDateFormat("EEEE,DD MMMM YYYY");
12
13          Tugas2_04 struk = new Tugas2_04(8);
14          Tugas2_04[] belanja = new Tugas2_04[8];
15          String date = sdf.format(waktu);
16
17          belanja[0] = new Tugas2_04("MEDSBK-090-445", 6, 2400000, date);
18          belanja[1] = new Tugas2_04("MEDSBK-080-446", 7, 1500000, date);
19          belanja[2] = new Tugas2_04("MEDSBK-056-447", 15, 1800000, date);
20          belanja[3] = new Tugas2_04("MEDSBK-683-448", 6, 190000, date);
21          belanja[4] = new Tugas2_04("MEDSBK-936-449", 2, 2356000, date);
22          belanja[5] = new Tugas2_04("MEDSBK-081-441", 5, 800000, date);
23          belanja[6] = new Tugas2_04("MEDSBK-482-442", 1, 6900000, date);
24          belanja[7] = new Tugas2_04("MEDSBK-593-443", 7, 4700000, date);
25
26          for (int i = 0; i < struk.size; i++) {
27              struk.push(belanja[i]);
28          }
29          struk.print();
30          System.out.println("\tData yang di-pop |struk 5 minggu|");
31          struk.line(50);
32          for (int i = 0; i < 5; i++) {
33              struk.pop();
34          }
35          struk.print();
36      }
37  }
```

Berikut merupakan hasil running dari kode program di atas

```
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

C:\Users\user\Documents\ASD 9>
```