

# **LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

## **PERTEMUAN KE 8**

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



Muhammad Afiq Firdaus

2341760189 / 21

SIB1E

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

## 7.2. Praktikum 1

Buat file Stack.java.

```
J Stack21.java > ...
1  public class Stack21 {
2      int data[];
3      int size;
4      int top;
5
6      public Stack21(int size){
7          this.size = size;
8          data = new int [size];
9          top = -1;
10     }
```

Lalu tambahkan method isFull() dan isEmpty() pada class Stack

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

Tambahkan method push(int data) dan pop()

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

ambahkan method peek()

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

Tambahkan method print() dan clear()

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

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

```
StackDemo21.java > StackDemo21
1  public class StackDemo21 {
2      public static void main(String[] args) {
3          Stack21 stack = new Stack21(10);
4          stack.push(8);
5          stack.push(12);
6          stack.push(18);
7          stack.print();
8          stack.pop();
9          stack.peek();
10         stack.push(-5);
11         stack.print();
12     }
13 }
```

## 7.2.2 Verifikasi Hasil Percobaan

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur D
tionMessages' '-cp' 'C:\Users\HP\AppData\Roaming\Code\User\work
1'
Isi stack :
18
12
8

Data yang dikeluarkan dari Stack : 18
Elemen teratas Stack: 12
Isi stack :
-5
12
8

PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur D
```

### 7.2.3 Pertanyaan

1. Pada method pop(), mengapa diperlukan pemanggilan method isEmpty()? Apa yang terjadi jika tidak ada pemanggilan isEmpty()?
2. Jelaskan perbedaan antara method peek() dengan method pop() pada class Stack.

#### Jawaban

1. Pemanggilan method isEmpty() pada method pop() diperlukan untuk memeriksa apakah stack sedang kosong sebelum mencoba mengeluarkan elemen dari stack. Hal ini penting karena jika stack kosong (tidak memiliki elemen), maka tidak ada elemen yang bisa di-"pop" (dikeluarkan). Jika tidak ada pemanggilan isEmpty(), maka ketika method pop() dipanggil pada stack kosong, proses pengambilan elemen akan dilakukan tanpa pengecekan. Ini dapat menyebabkan error atau situasi yang tidak diinginkan seperti mengakses indeks negatif dalam array data, yang dapat menyebabkan ArrayIndexOutOfBoundsException.
2. Dalam penggunaan stack, peek() dapat digunakan untuk melihat nilai elemen teratas sebelum memutuskan untuk menghapusnya menggunakan pop(). Keduanya adalah operasi penting dalam manipulasi data pada struktur data stack.

### 7.3. Praktikum 2

Buat class baru dengan nama Pakaian.

Tambahkan atribut-atribut Pakaian seperti pada Class Diagram Pakaian, kemudian tambahkan pula konstruktornya

```
Pakaian21.java > Pakaian21
1 public class Pakaian21 {
2     String jenis, warna, ukuran, merk;
3     double harga;
4
5     Pakaian21(String jenis, String warna, String ukuran, String merk, double harga){
6         this.jenis = jenis;
7         this.warna = warna;
8         this.ukuran = ukuran;
9         this.merk = merk;
10        this.harga = harga;
11    }
12 }
```

Setelah membuat class Pakaian, selanjutnya perlu dibuat class Stack yang berisi atribut dan method sesuai diagram Class Stack

tambahkan atribut dan konstruktor

Buat method isEmpty bertipe boolean yang digunakan untuk mengecek apakah stack kosong

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

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

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

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

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

Buat method clear bertipe void untuk menghapus seluruh isi stack.

```
Stack.java > Stack
1 public class Stack {
2     int size;
3     int top;
4     Pakaian21 data[];
5
6     public Stack(int size) {
7         this.size = size;
8         data = new Pakaian21[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 }
```

```
27
28 public void push(Pakaian21 pkn) {
29     if (!isFull()) {
30         top++;
31         data[top] = pkn;
32     } else {
33         System.out.println(x:"isi Stack penuh!");
34     }
35 }
36
37 public void pop() {
38     if (!isEmpty()) {
39         Pakaian21 x = data[top];
40         top--;
41         System.out.println(
42             "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 }
47
48 public void peek() {
49     System.out.println("Elemen teratas : " + data[top].jenis + " " + data[top].warna + " " + data[top].merk + " "
50         + data[top].ukuran + " " + data[top].harga);
51 }
52
53 public void print() {
54     System.out.println(x:"Isi Stack : ");
55     for (int i = top; i >= 0; i--) {
56         System.out.println(data[i].jenis + " " + data[i].warna + " " + data[i].merk + " " + data[i].ukuran + " "
57             + data[i].harga);
58     }
59 }
```

```

53  public void print() {
54      System.out.println(x:"Isi Stack : ");
55      for (int i = top; i >= 0; i--) {
56          System.out.println(data[i].jenis + " " + data[i].warna + " " + data[i].merk + " " + data[i].ukuran + " "
57              + data[i].harga);
58      }
59      System.out.println(x:" ");
60  }
61
62  public void clear() {
63      if (!isEmpty()) {
64          for (int i = top; i >= 0; i--) {
65              top--;
66          }
67          System.out.println(x:"Stack sudah dikosongkan ");
68      } else {
69          System.out.println(x:"Stack masih kosong");
70      }
71  }
72  }

```

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.

```

StackMain21.java > StackMain21 > main(String[])
1  import java.util.Scanner;
2  public class StackMain21 {
3      Run | Debug
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          Stack stk = new Stack(size:5);
7          char pilih;
8          do {
9              System.out.println(x:"Jenis: ");
10             String jenis = sc.nextLine();
11             System.out.println(x:"Warna: ");
12             String warna = sc.nextLine();
13             System.out.println(x:"Merk: ");
14             String merk = sc.nextLine();
15             System.out.println(x:"Ukuran: ");
16             String ukuran = sc.nextLine();
17             System.out.println(x:"Harga: ");
18             double harga = sc.nextDouble();
19
20             Pakaian21 p = new Pakaian21(jenis, warna, ukuran, merk, harga);
21             System.out.println(x:"Apakah Anda akan menambahkan data baru ke stack (y/n)? ");
22             pilih = sc.next().charAt(index:0);
23             sc.nextLine();
24             stk.push(p);
25
26             stk.print();
27             stk.pop();
28             stk.peek();
29             stk.print();
30         } while( pilih == 'y');
31     }
32 }

```



### 7.3.2. Verifikasi Hasil Percobaan

```

tikum\pertemuan 8'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' ^-XX:+ShowCo
ge\0fceeeca4cc3f50fe2019697f1491c13f\redhat.java\jdt_ws\pertemuan_8_a55985fd\bi
Jenis:
kaos
Warna:
hitam
Merk:
nevada
Ukuran:
m
Harga:
85000
Apakah Anda akan menambahkan data baru ke stack (y/n)?
y
Isi Stack:
kaos hitam nevada m 85000.0

Data yang keluar: kaos hitam nevada m 85000.0
Stack kosong, tidak ada elemen untuk diambil (peek)
Isi Stack:

Jenis:
kemeja
Warna:
putih
Merk:
styves
Ukuran:
xl
Harga:
127000
Apakah Anda akan menambahkan data baru ke stack (y/n)?

```

```

127000
Apakah Anda akan menambahkan data baru ke stack (y/n)?
y
Isi Stack:
kemeja putih styves xl 127000.0

Data yang keluar: kemeja putih styves xl 127000.0
Stack kosong, tidak ada elemen untuk diambil (peek)
Isi Stack:

Jenis:
calana
Warna:
biru
Merk:
levis
Ukuran:
l
Harga:
189500
Apakah Anda akan menambahkan data baru ke stack (y/n)?
n
Isi Stack:
calana biru levis l 189500.0

Data yang keluar: calana biru levis l 189500.0
Stack kosong, tidak ada elemen untuk diambil (peek)
Isi Stack:

```

PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 8> █

### 7.3.3. Pertanyaan

1. Berapa banyak data pakaian yang dapat ditampung di dalam stack? Tunjukkan potongan kode program untuk mendukung jawaban Anda tersebut!
2. Perhatikan class StackMain, pada saat memanggil fungsi push, parameter yang dikirimkan adalah p. Data apa yang tersimpan pada variabel p tersebut?
3. Apakah fungsi penggunaan do-while yang terdapat pada class StackMain?
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!

### Hasil

1. stack dapat menampung data pakaian sebanyak 5 buah karena Anda telah membuat objek Stack dengan ukuran 5 di dalam StackMain21
2. ketika memanggil fungsi push pada objek stk dengan parameter p yang merupakan objek Pakaian21, maka variabel p tersebut akan menyimpan data pakaian yang baru dimasukkan ke dalam stack.
3. ketika memanggil fungsi push pada objek stk dengan parameter p yang merupakan objek Pakaian21, maka variabel p tersebut akan menyimpan data pakaian yang baru dimasukkan ke dalam stack.

4.

```
32
33 import java.util.Scanner;
34 public class StackMain21 {
35     Run | Debug
36     public static void main (String args[]){
37         Scanner sc = new Scanner(System.in);
38         Stack stk = new Stack(size:5);
39         Scanner s = new Scanner (System.in);
40         char pilih;
41         boolean ulang = true;
42         do{
43             System.out.println(x:"*** MENU ***");
44             System.out.println(x:"1. PUSH");
45             System.out.println(x:"2. POP");
46             System.out.println(x:"3. PEEK");
47             System.out.println(x:"4. PRINT");
48             System.out.print(s:"Pilihanmu: ");
49             int angka = sc.nextInt();
50             switch(angka){
51                 case 1:
52                     boolean mengulang = true;
53                     do{
54                         System.out.print(s:"Jenis : ");
55                         String jenis = sc.next();
56                         System.out.print(s:"Warna : ");
57                         String warna = sc.next();
58                         System.out.print(s:"Merk :");
59                         String merk = sc.next();
60                         System.out.print(s:"Ukuran :");
61                         String ukuran = sc.next();
62                         System.out.print(s:"Harga :");
63                         double harga = sc.nextDouble();
64
```

```
65         Pakaian21 p = new Pakaian21 (jenis, warna, merk, ukuran, harga);
66         System.out.print(s:"Apakah Anda Akan Menambahkan Data Baru Ke Stack (y/n)? ");
67         pilih = sc.next().charAt(index:0);
68         sc.nextLine();
69         stk.push(p);
70     }while(pilih == 'y');
71     break;
72     case 2:
73         stk.pop();
74         System.out.print(s:"Apakah anda ingin kembali ke menu? (y/n): ");
75         pilih = sc.next().charAt(index:0);
76         sc.nextLine();
77         if (pilih == 'y'){
78             ulang = true;
79             break;
80         }
81     case 3:
82         stk.peek();
83         System.out.print(s:"Apakah anda ingin kembali ke menu? (y/n):");
84         pilih = sc.next().charAt(index:0);
85         sc.nextLine();
86         if (pilih == 'y'){
87             ulang = true;
88             break;
89         }

```



```

89         }
90     case 4:
91         stk.print();
92         System.out.print(s:"Apakah anda ingin kembali ke menu? (y/n): ");
93         pilih = sc.next().charAt(index:0);
94         sc.nextLine();
95         if (pilih == 'y'){
96             ulang = true;
97             break;
98         }
99     case 5:
100         ulang = false;
101         break;
102     }
103 }while (ulang);
104 }
105 }
106 // MODIFIKASI TUGAS PERCOBAAN 2 NOMOR 4

```

## 7.4. Praktikum 3

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

```

J PostFix21.java > PostFix21
1 public class PostFix21 {
2     int n, top;
3     char stack[];
4
5     public PostFix21(int total) {
6         n = total;
7         top = -1;
8         stack = new char[n];
9         push(c: '(');
10    }

```

Buat method push dan pop bertipe void.

```

11
12     public void push(char c) {
13         top++;
14         stack[top] = c;
15     }
16
17     public char pop() {
18         char item = stack[top];
19         top--;
20         return item;
21     }

```

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

```

23  public boolean IsOperand(char c) {
24      if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || (c == ' ' || c == '.')) {
25          return true;
26      } else {
27          return false;
28      }
29  }

```

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

```

31  public boolean IsOperator(char c) {
32      if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {
33          return true;
34      } else {
35          return false;
36      }
37  }

```

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

```

39  public int derajat(char c) {
40      switch (c) {
41          case '^':
42              return 3;
43          case '%':
44              return 2;
45          case '/':
46              return 2;
47          case '*':
48              return 2;
49          case '-':
50              return 1;
51          case '+':
52              return 1;
53          default:
54              return 0;
55      }
56  }

```

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

```

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

```

buat class baru dengan nama PostfixMain.

```

J PostFixMain21.java > PostFixMain21
1  import java.util.Scanner;
2  public class PostFixMain21 {
    Run | Debug
3      public static void main(String[] args){
4          Scanner sc = new Scanner(System.in);
5          String P, Q;
6          System.out.println(x:"Masukkan ekspresi matematika (infix): ");
7          Q = sc.nextLine();
8          Q = Q.trim();
9          Q = Q + ")";
10         int total = Q.length();
11         PostFix21 post = new PostFix21(total);
12         P = post.konversi(Q);
13         System.out.println("Postfix: " + P);
14     }
15 }

```

## 7.4.2. Verifikasi Hasil Percobaan

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 8> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -cp 'C:\Users\HVP\AppData\Roaming\Code\User\workspaceStorage\0fceeca4cc3f50fe2019697f1491c13f\redhat.java\jdt_ws\pertemuan 8_a55985fd\bin' 'PostFixMain'
Masukkan ekspresi matematika (infix):
a+b*(c+d-e)/f
Postfix: abcd+e-*f/+
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 8>
```

### 7.4.3. Pertanyaan

1. Perhatikan class Postfix, jelaskan alur kerja method derajat!
2. Apa fungsi kode program berikut?

```
c = Q.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!

#### Jawaban :

1. Pada dasarnya, method derajat(char c) melakukan pengecekan terhadap karakter c yang mewakili operator matematika (^, %, /, \*, -, +) dan mengembalikan nilai prioritas atau tingkat operator tersebut dalam konversi infix ke postfix. Prioritas ini digunakan untuk menentukan urutan pengambilan operator dalam konversi.
2. Kode program `c = Q.charAt(i);` memiliki fungsi untuk mengambil karakter pada posisi i dari string Q dan kemudian menyimpan karakter tersebut ke dalam variabel c.

```
ge\0fceeca4cc3f50fe2019697f1491c13f\redhat.java\jdt_ws\pertemuan 8_a55985fd\bin' 'PostFixMain'
Masukkan ekspresi matematika (infix):
5*4^(1+2)%3
Postfix: 5412+^*3%
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 8>
```

4. Dalam hasil konversi dari ekspresi matematika infix ke postfix menggunakan algoritma yang diimplementasikan dalam program PostFix21, tanda kurung ( dan ) tidak akan muncul dalam hasil postfix yang dihasilkan. Hal ini terjadi karena tanda kurung hanya digunakan sebagai alat untuk mengatur urutan operasi dalam ekspresi matematika infix, namun tidak termasuk sebagai bagian dari operasi matematika pada 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!

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!

**Jawaban :**

1.

```
79 void getMax() {
80     if (!IsEmpty()) {
81         Pakaian21 maxPakaian = data[0];
82         for (int i = 1; i <= top; i++) {
83             if (data[i].harga > maxPakaian.harga) {
84                 maxPakaian = data[i];
85             }
86         }
87         System.out.println("Pakaian dengan harga tertinggi: " + maxPakaian.jenis + " " + maxPakaian.warna + " "
88                             + maxPakaian.merk + " " + maxPakaian.ukuran + " " + maxPakaian.harga);
89     } else {
90         System.out.println(x:"Stack kosong, tidak ada pakaian untuk diperiksa harga tertinggi");
91     }
92 }
93 }
```

2.

```
1 public class StrukBelanja {
2     String nomorTransaksi;
3     String tanggalPembelian;
4     int jumlahBarang;
5     int totalHargaBayar;
6
7     public StrukBelanja(String nomorTransaksi, String tanggalPembelian, int jumlahBarang, int totalHargaBayar) {
8         this.nomorTransaksi = nomorTransaksi;
9         this.tanggalPembelian = tanggalPembelian;
10        this.jumlahBarang = jumlahBarang;
11        this.totalHargaBayar = totalHargaBayar;
12    }
13 }
14 }
```

```

1  import java.util.Stack;
2
3  public class StrukBelanjaMain {
4      public static void main(String[] args) {
5          Stack<StrukBelanja> stackStrukBelanja = new Stack<>();
6
7          // Menambahkan struk belanja ke dalam stack
8          stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"001", tanggalPembelian:"2024-04-01", jumlahBaran...5, 150000));
9          stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"002", tanggalPembelian:"2024-04-08", jumlahBaran...3, 90000));
10         stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"003", tanggalPembelian:"2024-04-15", jumlahBaran...4, 120000));
11         stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"004", tanggalPembelian:"2024-04-22", jumlahBaran...6, 180000));
12         stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"005", tanggalPembelian:"2024-04-29", jumlahBaran...2, 60000));
13         stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"006", tanggalPembelian:"2024-05-06", jumlahBaran...7, 210000));
14         stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"007", tanggalPembelian:"2024-05-13", jumlahBaran...4, 120000));
15         stackStrukBelanja.push(new StrukBelanja(nomorTransaksi:"008", tanggalPembelian:"2024-05-20", jumlahBaran...3, 90000));
16
17         // Menampilkan informasi struk belanja yang masih tersimpan di dalam stack
18         System.out.println(x:"Informasi Struk Belanja yang Masih Tersimpan:");
19         while (!stackStrukBelanja.isEmpty()) {
20             StrukBelanja struk = stackStrukBelanja.pop();
21             System.out.println("Nomor Transaksi: " + struk.nomorTransaksi);
22             System.out.println("Tanggal Pembelian: " + struk.tanggalPembelian);
23             System.out.println("Jumlah Barang: " + struk.jumlahBarang);
24             System.out.println("Total Harga Bayar: " + struk.totalHargaBayar);
25             System.out.println();
26         }
27     }
28 }

```