**LAPORAN PRAKTIKUM JOBSHEET 8**

**STACK**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**



**Disusun Oleh :**

**Jami’atul Afifah (2341760102)**

**SIB-1F**

**PROGRAM STUDI D4 SISTEM INFOEMASI BISNIS**

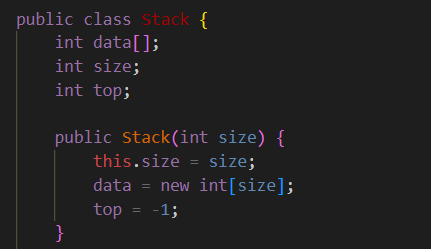
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

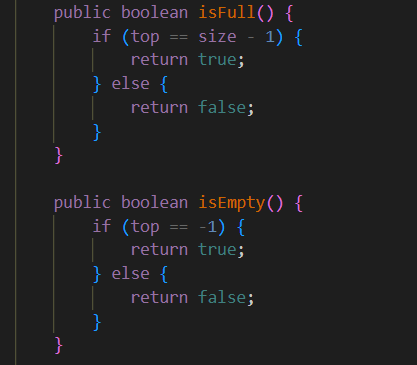
**2024**

**Praktikum 1**

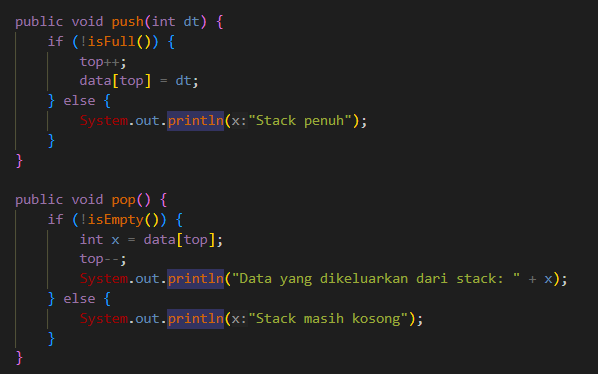
1. Buat folder dengan nama Praktikum07. Buat file Stack.java.
2. Tulis kode untuk membuat atribut dan konstruktor pada class Stack sebagai berikut:



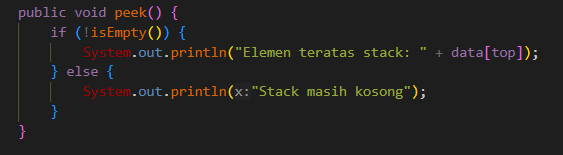
1. Lalu tambahkan method isFull() dan isEmpty() pada class Stack sebagai berikut:



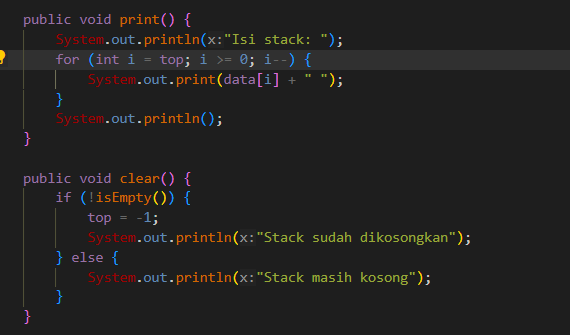
1. Tambahkan method push(int data) dan pop() sebagai berikut:



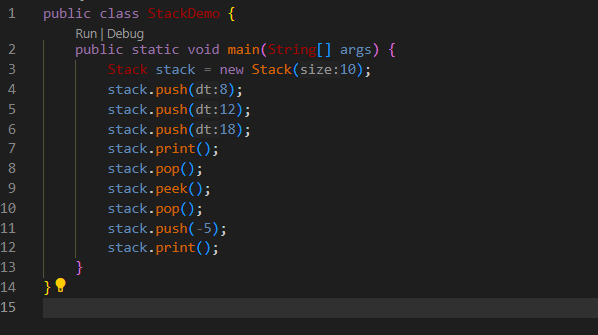
1. Tambahkan method peek() sebagai berikut:



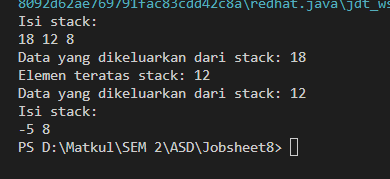
1. Tambahkan method print() dan clear() sebagai berikut:



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



1. Compile dan run class StackDemo.



**Pertanyaan**

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

Pemanggilan method isEmpty() pada method pop() diperlukan untuk memastikan bahwa stack tidak kosong sebelum mencoba untuk menghapus elemen dari stack. Jika tidak ada pemanggilan isEmpty(), maka pada saat method pop() dipanggil, program akan mencoba untuk menghapus elemen dari stack tanpa memeriksa apakah stack tersebut kosong atau tidak. Hal ini dapat menyebabkan error atau kegagalan dalam program, terutama jika stack sudah kosong dan kita mencoba untuk menghapus elemen dari stack yang tidak memiliki elemen. Dengan memeriksa apakah stack kosong sebelum melakukan operasi pop, kita dapat menghindari kesalahan seperti itu dan menangani kasus stack kosong secara lebih aman.

1. Jelaskan perbedaan antara method peek() dengan method pop() pada class Stack.
   1. peek(): Method peek() digunakan untuk melihat elemen teratas dari stack tanpa menghapusnya dari stack. Ini berarti bahwa elemen teratas tetap berada di stack setelah pemanggilan peek(). Method peek() hanya mengembalikan nilai dari elemen teratas tanpa mengubah keadaan stack.
   2. pop(): Method pop() digunakan untuk menghapus elemen teratas dari stack dan mengembalikan nilainya. Setelah pemanggilan pop(), elemen teratas stack dihapus dan stack akan menjadi lebih pendek. Dengan kata lain, pop() tidak hanya mengembalikan nilai dari elemen teratas, tetapi juga mengubah keadaan stack dengan menghapus elemen teratasnya.

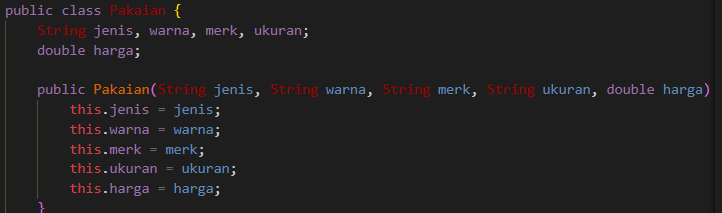
**Praktikum 2**

1. Perhatikan Diagram Class Pakaian berikut ini:

|  |
| --- |
| Pakaian |
| jenis: String warna: String merk: String ukuran: String  harga: double |
| Pakaian(jenis: String, warna: String, merk: String, ukuran: String, harga: double) |

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

1. Buat class baru dengan nama Pakaian.
2. Tambahkan atribut-atribut Pakaian seperti pada Class Diagram Pakaian, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.

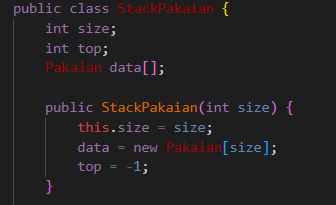


1. Setelah membuat class Pakaian, selanjutnya perlu dibuat class Stack yang berisi atribut dan method sesuai diagram Class Stack berikut ini:

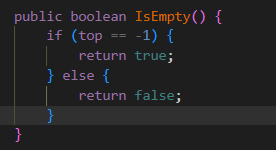
|  |
| --- |
| Stack |
| size: int top: int  data[]: Pakaian |
| Stack(size: int) IsEmpty(): boolean IsFull(): boolean  push(): void  pop(): void  peek(): void  print(): void  clear(): void |

Keterangan: Tipe data pada variabell data menyesuaikan dengan data yang akan 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

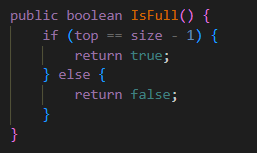
1. Buat class baru dengan nama Stack. Kemudian tambahkan atribut dan konstruktor seperti gambar berikut ini.



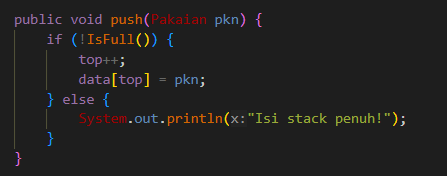
1. Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah stack kosong.



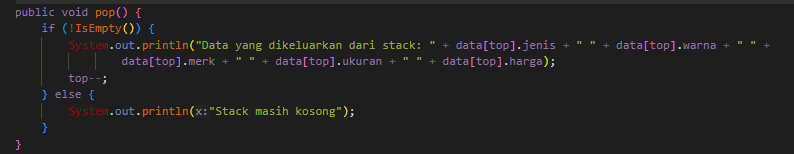
1. Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah stack sudah terisi penuh.



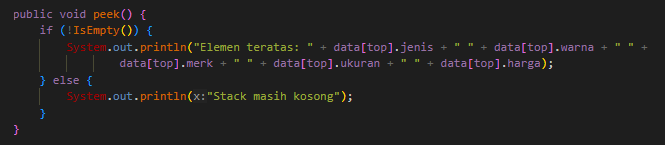
1. Buat method push bertipe void untuk menambahkan isi elemen stack dengan parameter pkn



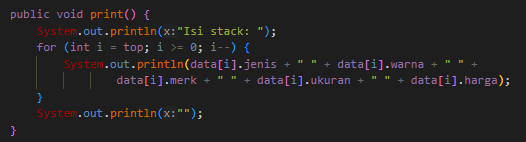
1. Buat method Pop bertipe void untuk mengeluarkan isi elemen stack. Karena satu elemen stack terdiri dari beberapa informasi (jenis, warna, merk, ukuran, dan harga), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut



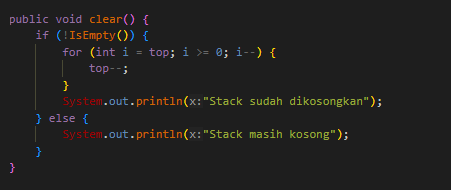
1. Buat method peek bertipe void untuk memeriksa elemen stack pada posisi paling atas.



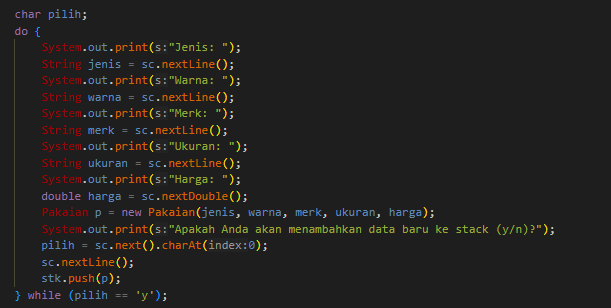
1. Buat method print bertipe void untuk menampilkan seluruh elemen pada stack.



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

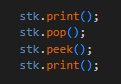


1. Selanjutnya, buat class baru dengan nama StackMain. Buat fungsi main, kemudian lakukan instansiasi objek dari class Stack dengan nama stk dan nilai parameternya adalah Stack stk = new Stack(5);
2. Deklarasikan Scanner dengan nama sc
3. Tambahkan kode berikut ini untuk menerima input data Pakaian, kemudian semua informasi tersebut dimasukkan ke dalam stack

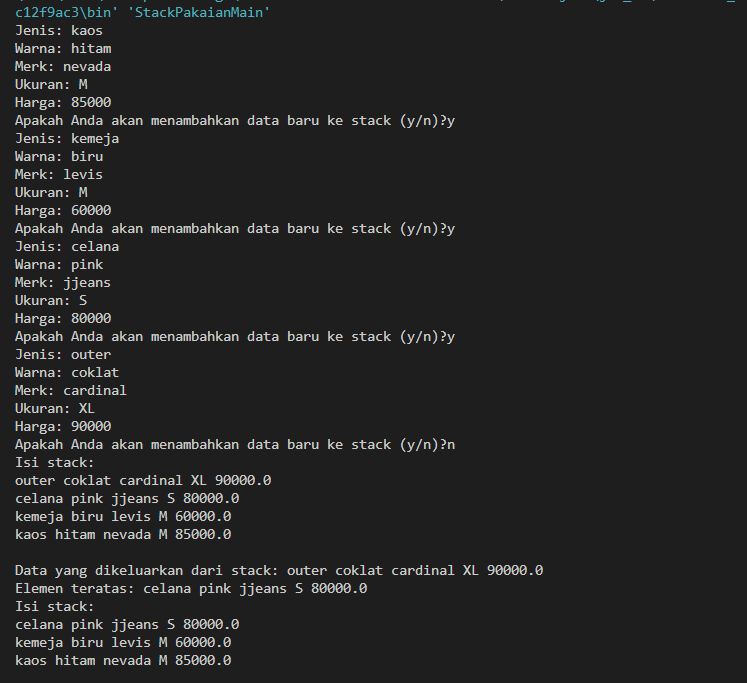


Catatan: sintaks sc.nextLine() sebelum sintaks st.push(p) digunakan untuk mengabaikan karakter new line

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



1. Compile dan jalankan class StackMain, kemudian amati hasilnya



**Pertanyaan**

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

Jumlah data pakaian yang dapat ditampung di dalam stack ditentukan oleh ukuran stack yang didefinisikan pada saat pembuatan objek stack. Dalam kasus ini, ukuran stack ditentukan saat pembuatan objek StackPakaian. Sebagai contoh, jika kita membuat objek StackPakaian dengan ukuran 5, maka stack tersebut dapat menampung hingga 5 data pakaian.



1. Perhatikan class StackMain, pada saat memanggil fungsi push, parameter yang dikirimkan adalah p. Data apa yang tersimpan pada variabel p tersebut?

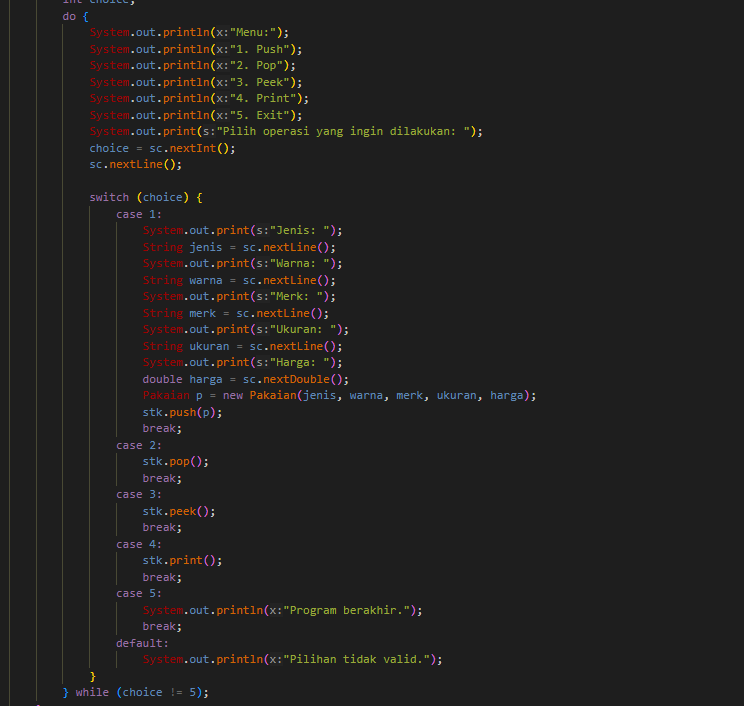


Pada saat memanggil fungsi push dalam class StackMain, parameter yang dikirimkan adalah objek p yang merupakan instansiasi dari kelas Pakaian. Data yang tersimpan pada variabel p adalah informasi tentang pakaian yang dimasukkan oleh pengguna, yaitu jenis, warna, merk, ukuran, dan harga.

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

Fungsi penggunaan do-while pada class StackMain digunakan untuk meminta input dari pengguna untuk menambahkan data pakaian ke dalam stack. Program akan terus meminta input hingga pengguna memilih untuk tidak menambahkan data lagi (memasukkan input selain 'y'). Ini memungkinkan pengguna untuk menambahkan lebih dari satu data pakaian secara berturut-turut.

1. 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!

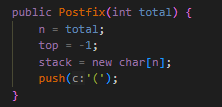


**Praktikum 3**

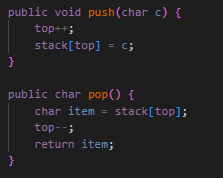
1. Perhatikan Diagram Class berikut ini:

|  |
| --- |
| 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 |

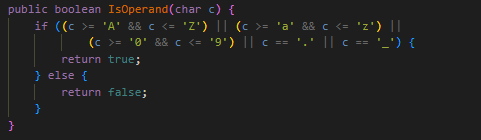
1. Buat class baru dengan nama Postfix. Tambahkan atribut n, top, dan stack sesuai diagram class Postfix tersebut.
2. Tambahkan pula konstruktor berparameter seperti gambar berikut ini.



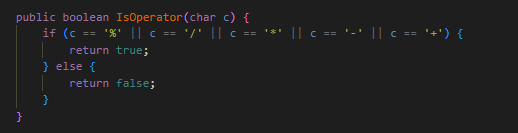
1. Buat method push dan pop bertipe void.



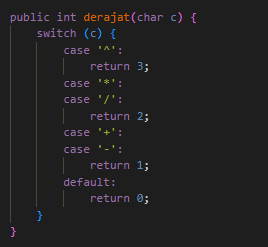
1. Buat method IsOperand dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operand.



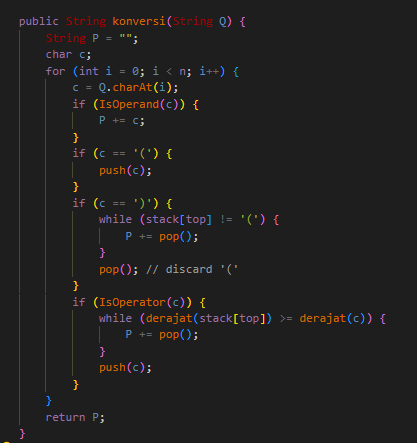
1. Buat method IsOperator dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operator.



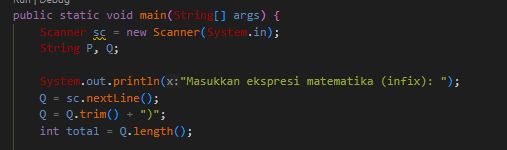
1. Buat method derajat yang mempunyai nilai kembalian integer untuk menentukan derajat operator.



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

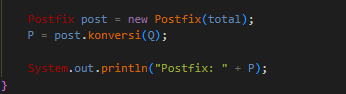


1. Selanjutnya, buat class baru dengan nama PostfixMain. Buat class main, kemudian buat variabel P dan Q. Variabel P digunakan untuk menyimpan hasil akhir notasi postfix setelah dikonversi, sedangkan variabel Q digunakan untuk menyimpan masukan dari pengguna berupa ekspresi matematika dengan notasi infix. 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.

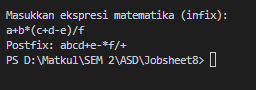


Penambahan string “)” digunakan untuk memastikan semua simbol/karakter yang masih berada di stack setelah semua persamaan terbaca, akan dikeluarkan dan dipindahkan ke postfix.

1. Buat variabel total untuk menghitung banyaknya karaketer pada variabel Q.
2. 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. Compile dan jalankan class PostfixMain dan amati hasilnya.



**Pertanyaan**

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

Method derajat(char c) pada class Postfix berfungsi untuk menentukan derajat operator matematika. Derajat operator digunakan untuk menentukan urutan operasi dalam ekspresi matematika, di mana operator dengan derajat yang lebih tinggi akan dievaluasi terlebih dahulu. Alur kerja method derajat adalah sebagai berikut:

1. Method menerima satu karakter operator sebagai argumen.
2. Kemudian, menggunakan sebuah switch-case, method memeriksa karakter tersebut dan mengembalikan nilai derajat sesuai dengan operator tersebut.
3. Untuk operator ^ (pangkat), derajatnya adalah 3.
4. Untuk operator \* dan / (perkalian dan pembagian), derajatnya adalah 2.
5. Untuk operator + dan - (penjumlahan dan pengurangan), derajatnya adalah 1.
6. Jika karakter tidak cocok dengan operator apa pun, method mengembalikan nilai 0.
7. Apa fungsi kode program berikut?

Kode program c = Q.charAt(i); berfungsi untuk mengambil karakter pada indeks ke-i dari string Q dan menyimpannya ke dalam variabel c. Dalam konteks ini, string Q adalah ekspresi matematika dalam notasi infix, dan variabel c digunakan untuk mewakili karakter saat iterasi melalui string Q.

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



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

Tanda kurung "(" dan ")" tidak ditampilkan pada hasil konversi karena dalam algoritma konversi dari infix ke postfix, tanda kurung hanya digunakan sebagai pembatas untuk mengatur urutan operasi. Ketika menemui tanda kurung buka "(", maka tanda kurung tersebut akan didorong ke dalam stack, dan ketika menemui tanda kurung tutup ")", maka operator-operator yang ada dalam stack akan dikeluarkan hingga menemui tanda kurung buka yang sesuai. Tanda kurung tersebut tidak termasuk dalam ekspresi postfix karena hanya digunakan untuk mengatur urutan operasi dan tidak menyatakan operasi matematika itu sendiri. Oleh karena itu, tanda kurung tidak ditampilkan pada hasil konversi.

**Tugas**

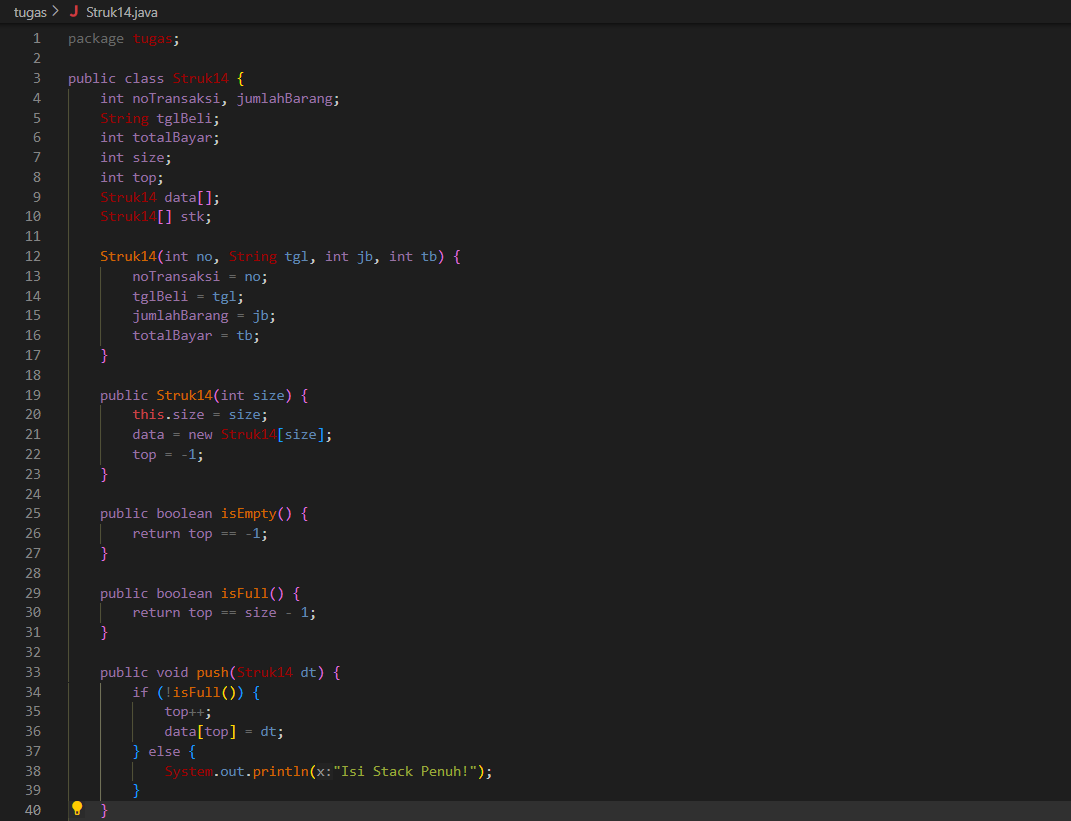
1. Tambahkan method getMax pada class Stack yang digunakan untuk mencari dan menampilkan data pakaian dengan harga tertinggi dari semua data pakaian yang tersimpan di dalam stack!

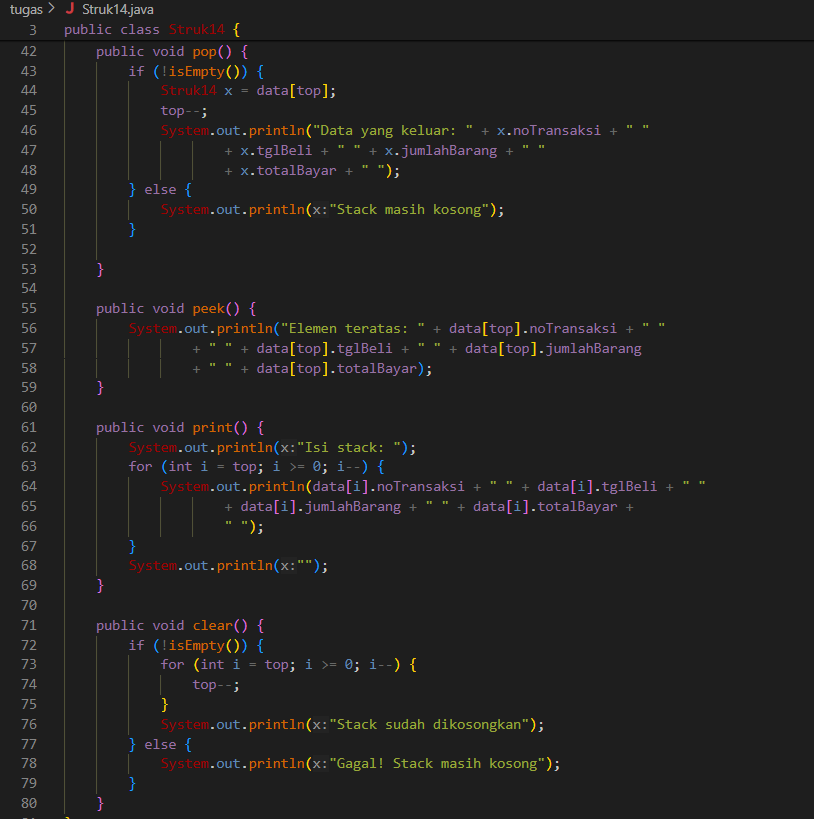


1. Setiap hari Minggu, Dewi pergi berbelanja ke salah satu supermarket yang berada di area rumahnya. Setiap kali selesai berbelanja, Dewi menyimpan struk belanjaannya di dalam laci. Setelah dua bulan, ternyata Dewi sudah mempunyai delapan struk belanja. Dewi berencana mengambil lima struk belanja untuk 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:
   1. Nomor transaksi
   2. Tanggal pembelian
   3. Jumlah barang yang dibeli
   4. Total harga bayar

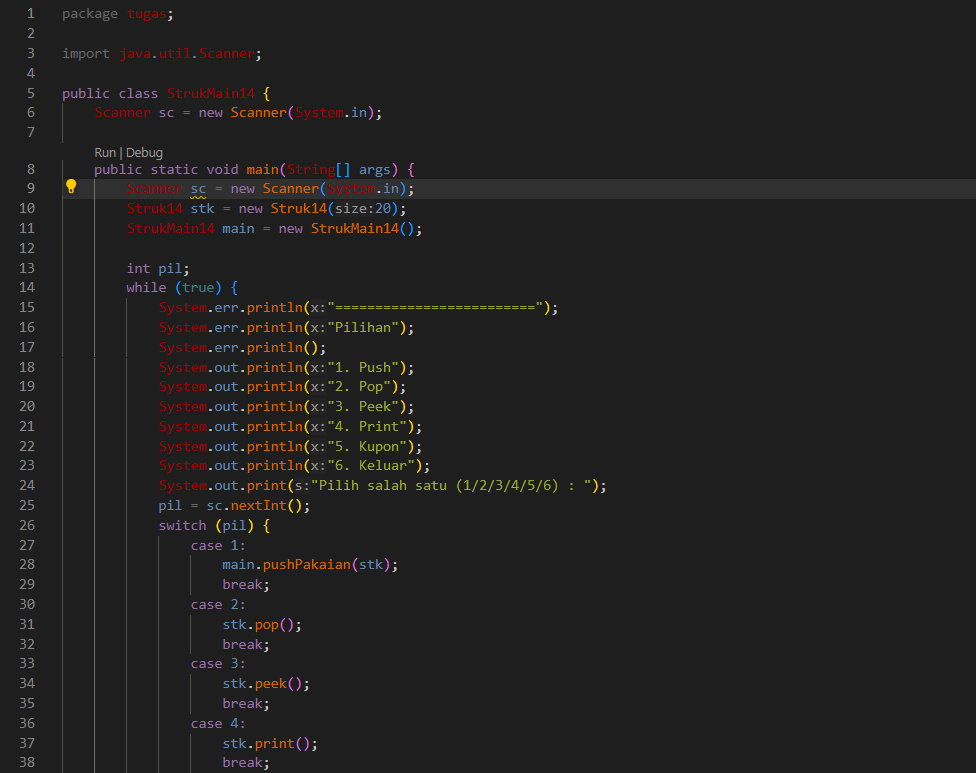
Tampilkan informasi struk belanja yang masih tersimpan di dalam stack

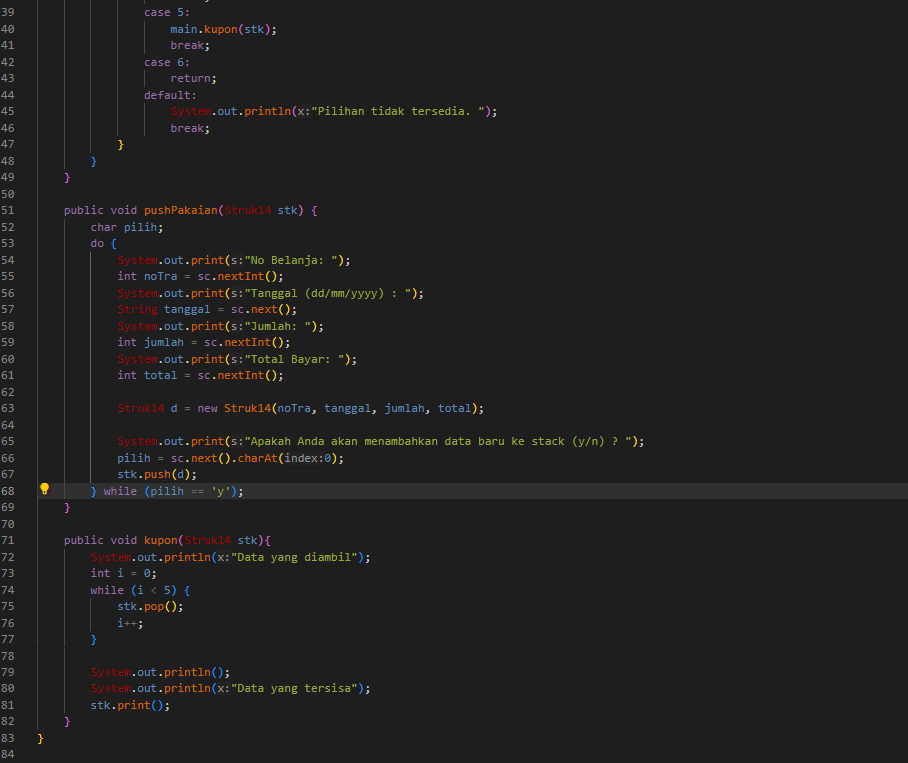
**Buat File Struk17.java**





**Buat File StrukMain.java**





**Hasil:**

