**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**

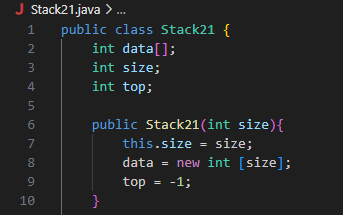
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

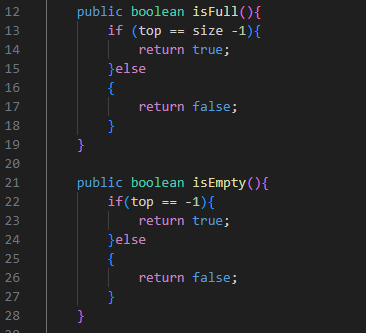
**2024**

**7.2. Praktikum 1**

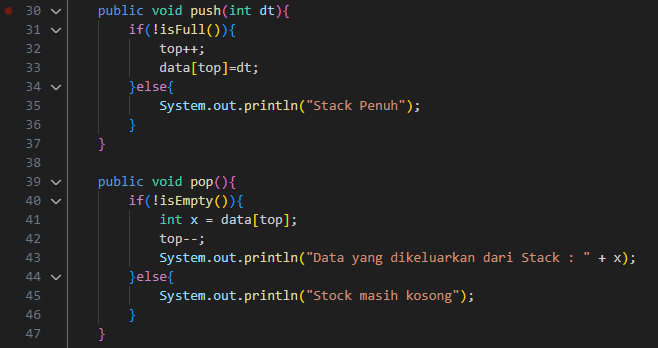
Buat file Stack.java.



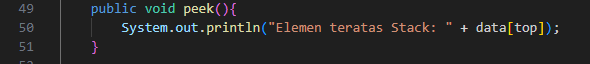
Lalu tambahkan method isFull() dan isEmpty() pada class Stack



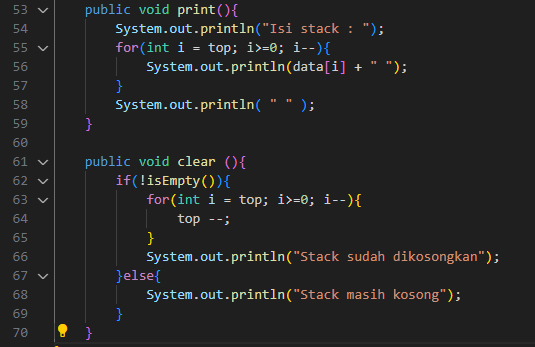
Tambahkan method push(int data) dan pop()



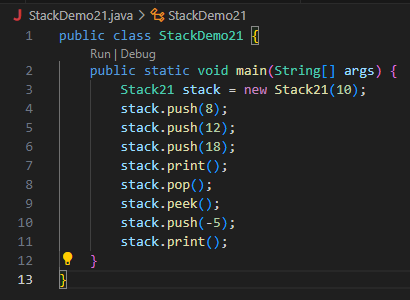
ambahkan method peek()



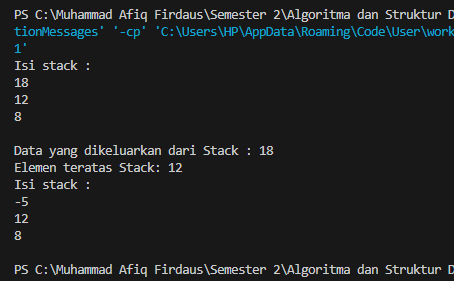
Tambahkan method print() dan clear()



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



**7.2.2 Verifikasi Hasil Percobaan**

****

**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.

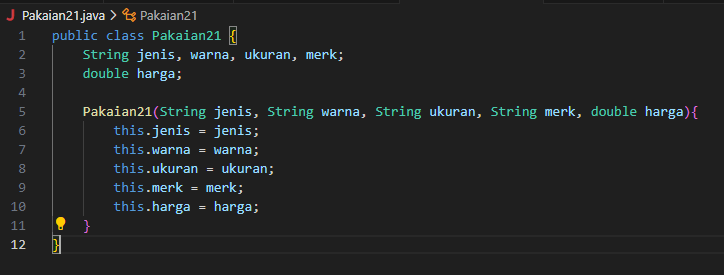
**Jawaban**

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



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.

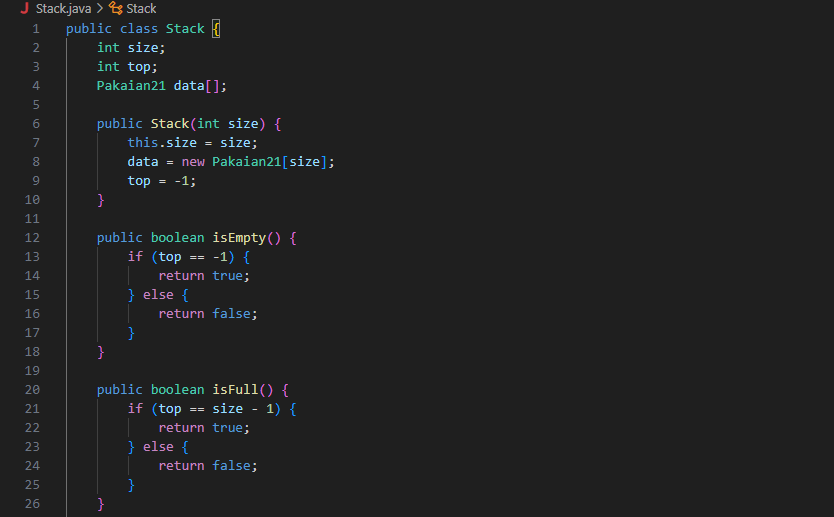
Buat method push bertipe void untuk menambahkan isi elemen stack dengan parameter pkn yang berupa object Pakaian

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

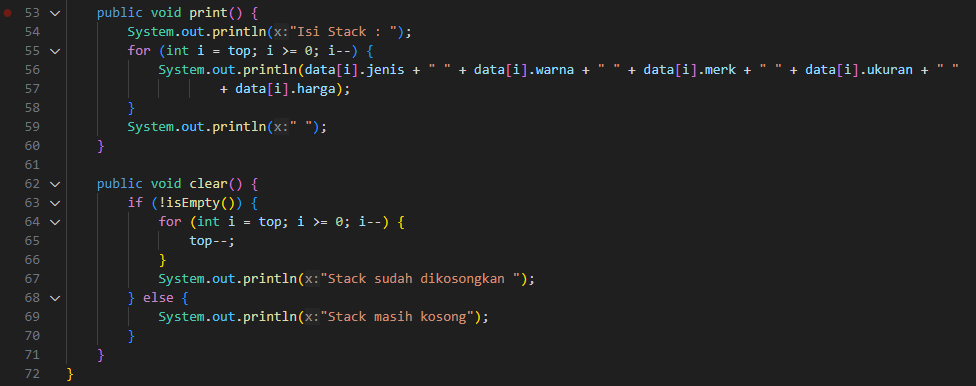
Buat method peek bertipe void untuk memeriksa elemen stack pada posisi paling atas

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

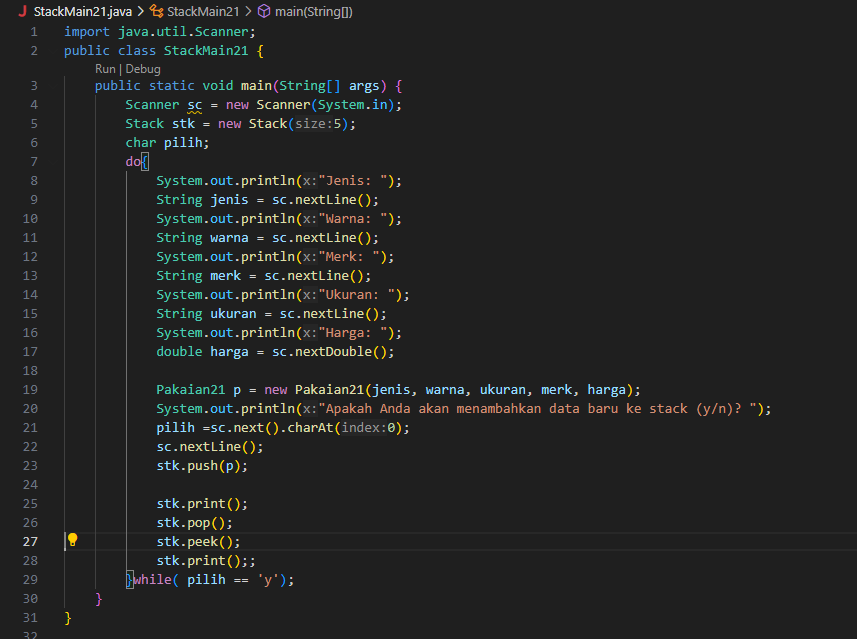
Buat method clear bertipe void untuk menghapus seluruh isi stack.



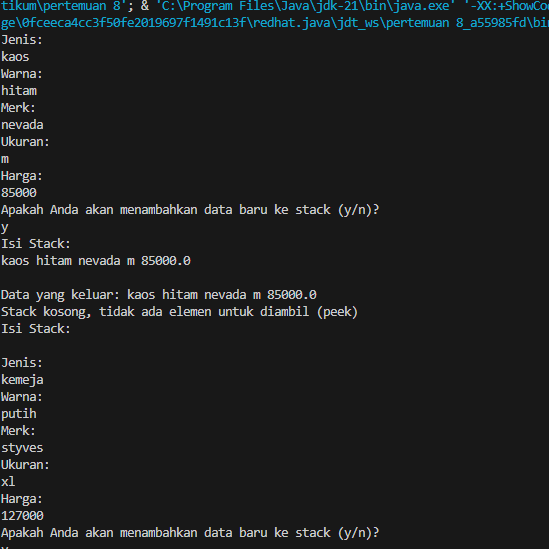


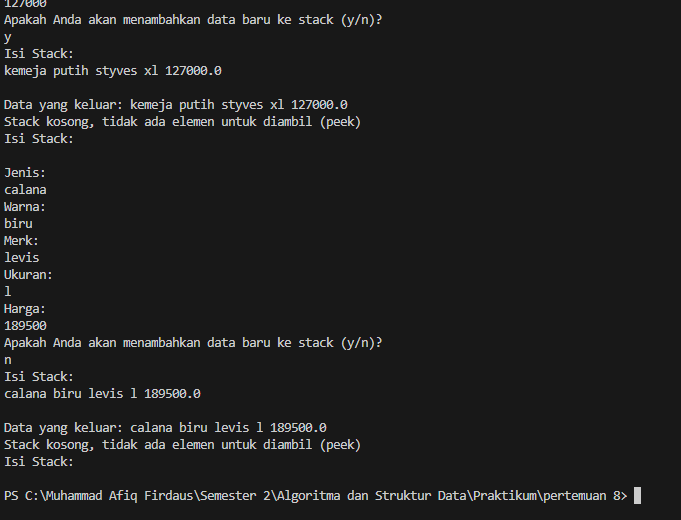


Selanjutnya, buat class baru dengan nama StackMain. Buat fungsi main, kemudian lakukan instansiasi objek dari class Stack dengan nama stk dan nilai parameternya adalah 5.



**7.3.2. Verifikasi Hasil Percobaan**





**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?

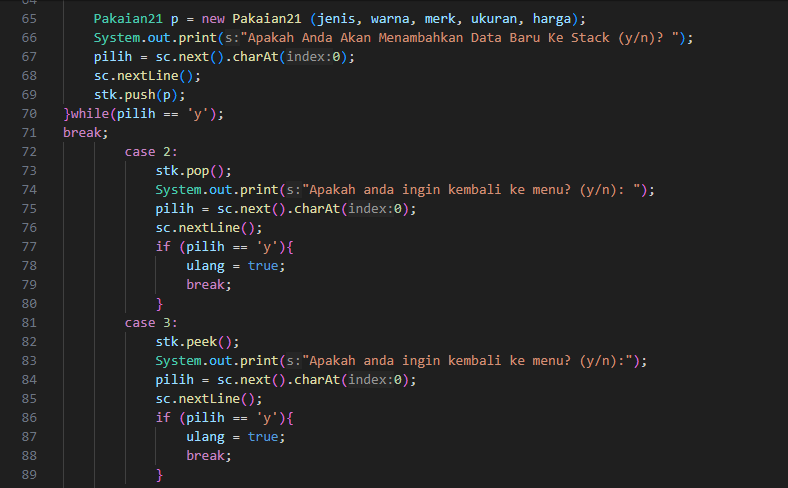
4. Modifikasi kode program pada class StackMain sehingga pengguna dapat memilih operasioperasi pada stack (push, pop, peek, atau print) melalui pilihan menu program dengan memanfaatkan kondisi IF-ELSE atau SWITCH-CASE!

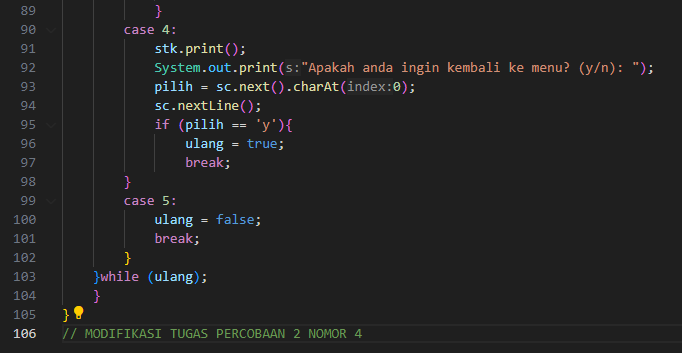
**Hasil**1. stack dapat menampung data pakaian sebanyak 5 buah karena Anda telah membuat objek Stack dengan ukuran 5 di dalam StackMain21

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.

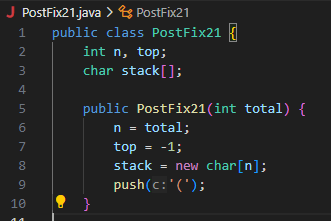
4. 



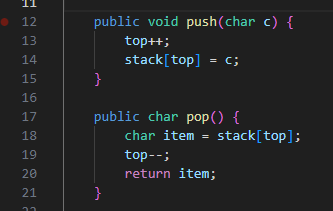


**7.4. Praktikum 3**

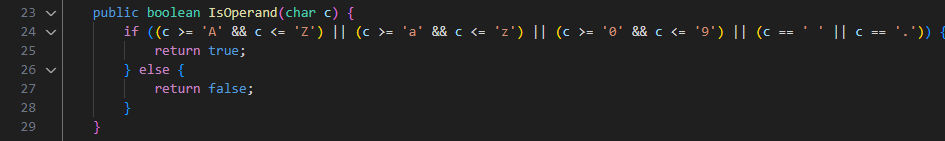
Buat class baru dengan nama Postfix. Tambahkan atribut n, top, dan stack sesuai diagram class Postfix



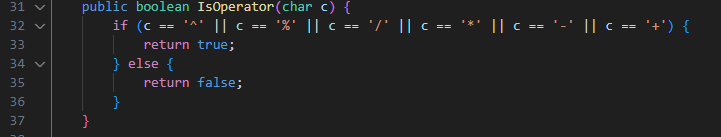
Buat method push dan pop bertipe void.



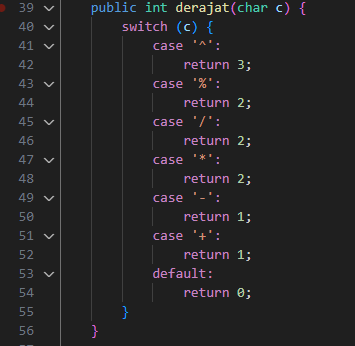
method IsOperand dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operand.



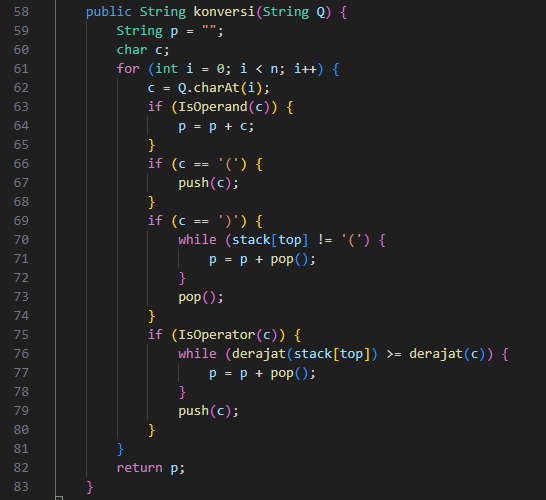
method IsOperator dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operator



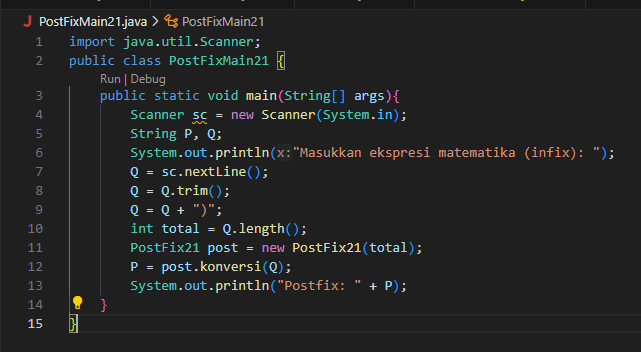
method derajat yang mempunyai nilai kembalian integer untuk menentukan derajat operator.



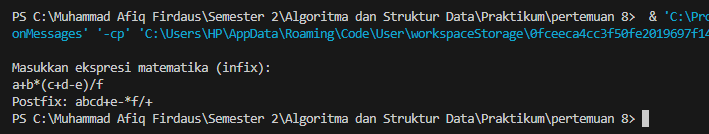
Buat method konversi untuk melakukan konversi notasi infix menjadi notasi postfix dengan cara mengecek satu persatu elemen data pada String Q sebagai parameter masukan.



buat class baru dengan nama PostfixMain.



**7.4.2. Verifikasi Hasil Percobaan**



**7.4.3. Pertanyaan**

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

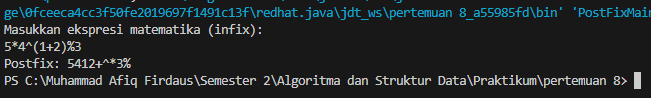
2. Apa fungsi kode program berikut?



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.
3. 
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 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:

• Nomor transaksi

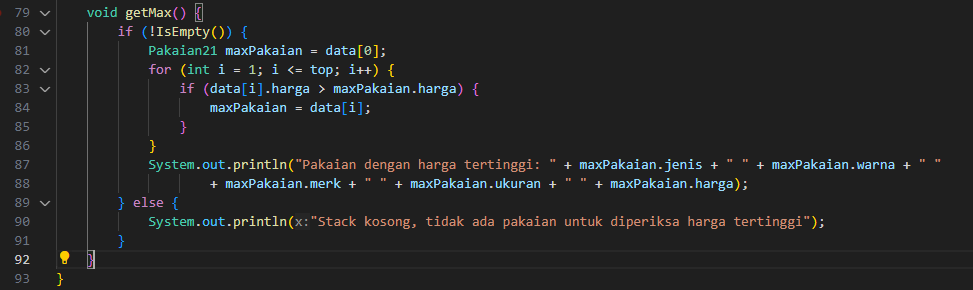
• Tanggal pembelian

• Jumlah barang yang dibeli

• Total harga bayar

Tampilkan informasi struk belanja yang masih tersimpan di dalam stack!

**Jawaban :**

1. 
2. 