**LAPORAN PRAKTIKUM**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**

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

**PERTEMUAN - 7 - Searching**

****

**Nama : M. Zidna Billah Faza**

**NIM : 2341760030**

**Prodi : D-IV Sistem Informasi Bisnis**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

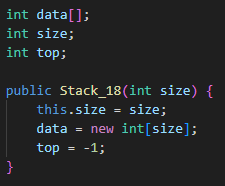
**2024**

**Percobaan 1**

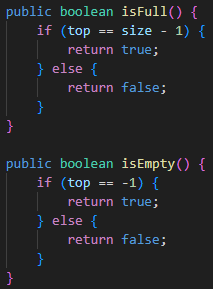
1. Buat folder dengan nama Praktikum07. Buat file Stack.java.



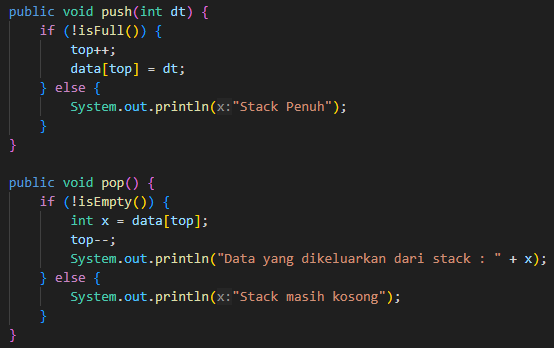
1. Tulis kode untuk membuat atribut dan konstruktor pada class Stack sebagai berikut:



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



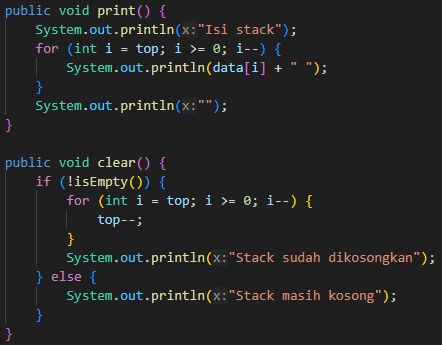
1. Tambahkan method push(int data) dan pop() sebagai berikut:



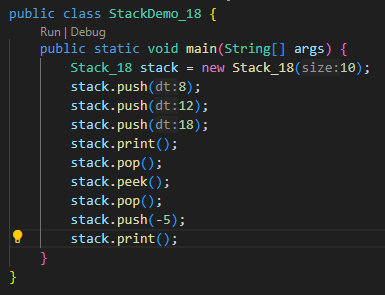
1. Tambahkan method peek() sebagai berikut:



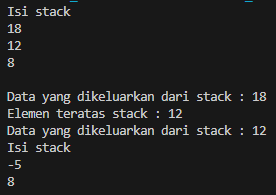
1. Tambahkan method print() dan clear() sebagai berikut:



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 dan verifikasi hasil percobaan



Pertanyaan Percobaan 1

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

Untuk menjamin operasi yang aman pada struktur data dan menghindari pengecualian dalam metode pop(), maka metode isEmpty() harus dipanggil. Ketika isEmpty tidak dipanggil maka akan ada pengecualian.

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

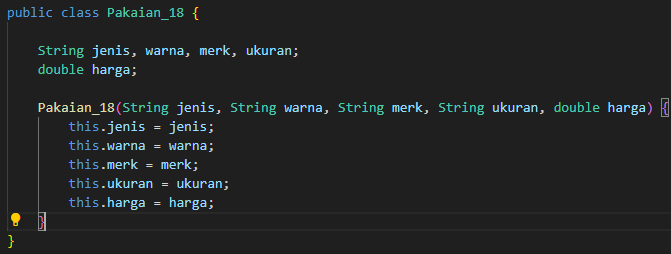
* peek() : digunakan untuk melihat elemen teratas pada stack atau top tanpa menghapusnya
* pop() : digunakan untuk menghapus elemen teratas atau top

**Percobaan 2**

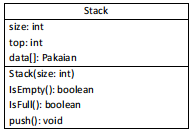
1. Buat class baru dengan nama Pakaian



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



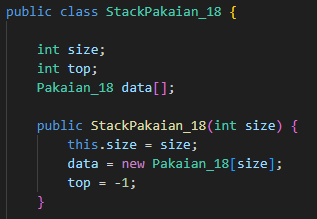
1. Setelah membuat class Pakaian, selanjutnya perlu dibuat class Stack yang berisi atribut dan method sesuai diagram Class Stack berikut ini:



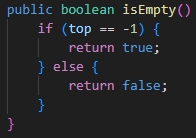


Keterangan: Tipe data pada variabell data menyesuaikan dengan data yang akan akan disimpan di dalam Stack. Pada praktikum ini, data yang akan disimpan merupakan array of object dari Pakaian, sehingga tipe data yang digunakan adalah Pakaian

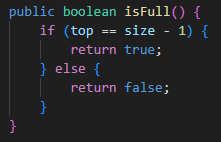
1. Buat class baru dengan nama Stack. Kemudian tambahkan atribut dan konstruktor seperti gambar berikut ini.



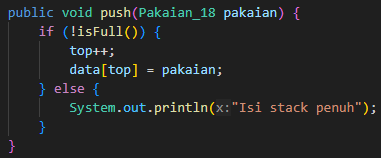
1. Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah stack kosong.



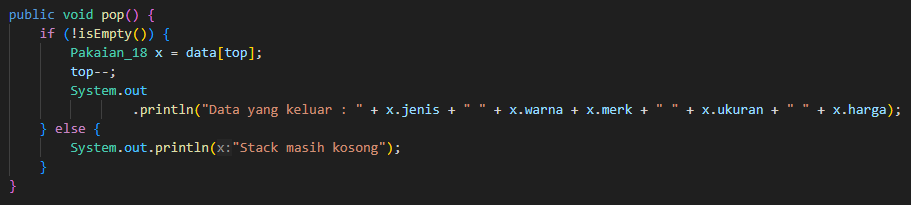
1. Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah stack sudah terisi penuh.



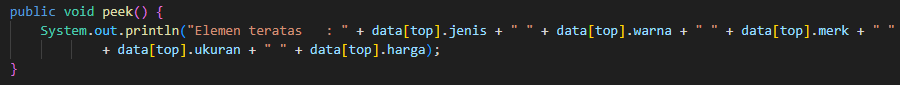
1. Buat method push bertipe void untuk menambahkan isi elemen stack dengan parameter pakaian yang berupa object pakaian



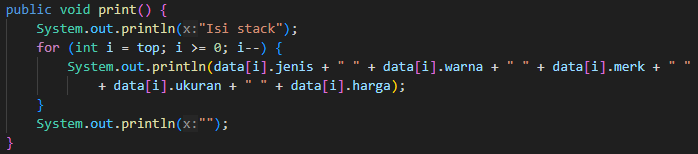
1. Buat method Pop bertipe void untuk mengeluarkan isi elemen stack. Karena satu elemen stack terdiri dari beberapa informasi (jenis, warna, merk, ukuran, dan harga), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut



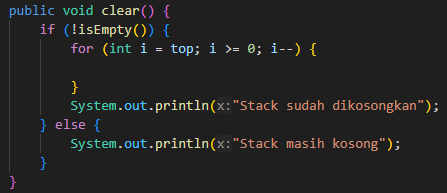
1. Buat method peek bertipe void untuk memeriksa elemen stack pada posisi paling atas.



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



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



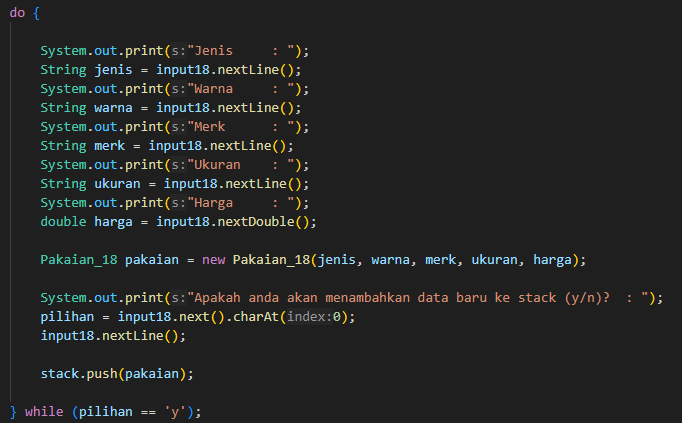
1. Selanjutnya, buat class baru dengan nama StackMain. Buat fungsi main, kemudian lakukan instansiasi objek dari class Stack dengan nama stack dan nilai parameternya adalah 5.



1. Deklarasikan Scanner dengan nama input18

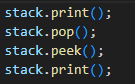


1. Tambahkan kode berikut ini untuk menerima input data Pakaian, kemudian semua informasi tersebut dimasukkan ke dalam stack

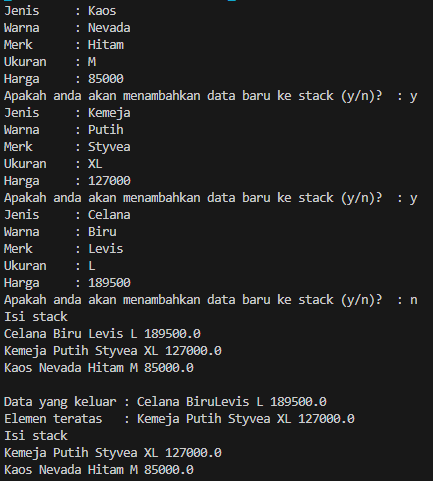


Catatan: sintaks sc.nextLine() sebelum sintaks st.push(p) digunakan untuk mengabaikan karakter new line

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



1. Compile dan jalankan class StackMain, kemudian amati hasilnya serta verifikasi hasil percobaan.



**Pertanyaan Percobaan 2**

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

Terdapat 5 data yang dapat ditampung pada stack



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

Data yang tersimpan adalah data objek dengan atribut jenis, merk, ukuran, dan harga





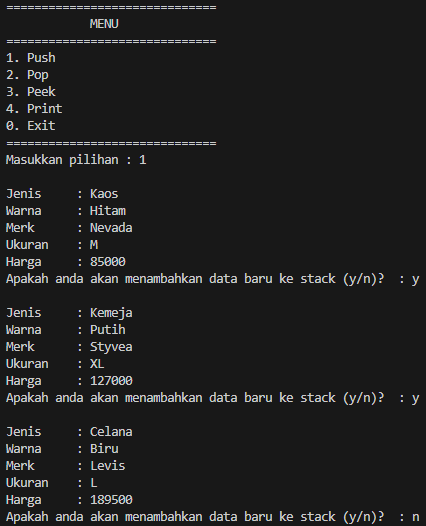
Catatan p = pakaian

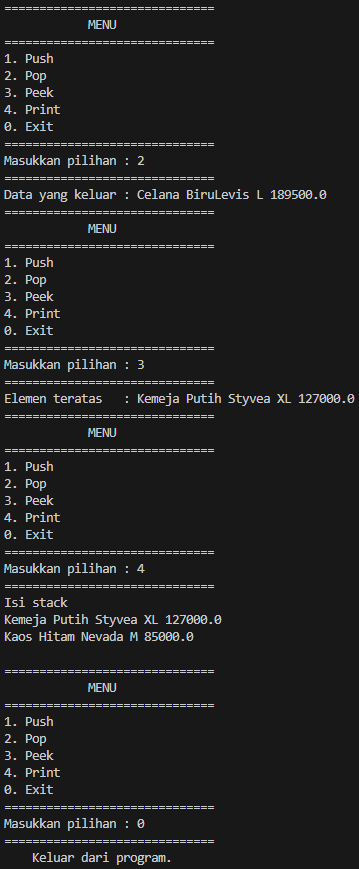
1. Apakah fungsi penggunaan do-while yang terdapat pada class StackMain?

Untuk perulangan apabila user ingin memasukkan data baru

1. 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!





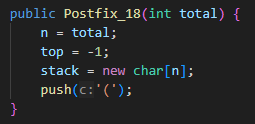


**Percobaan 3**

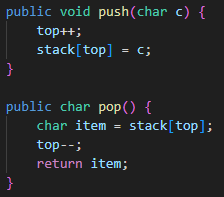
1. Buat class baru dengan nama Postfix. Tambahkan atribut n, top, dan stack sesuai diagram class Postfix tersebut.



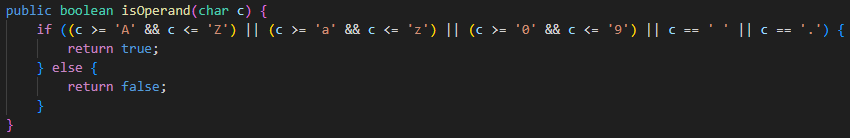
1. Tambahkan pula konstruktor berparameter seperti gambar berikut ini.



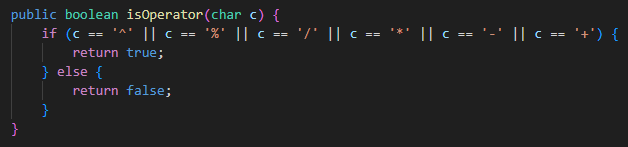
1. Buat method push dan pop bertipe void



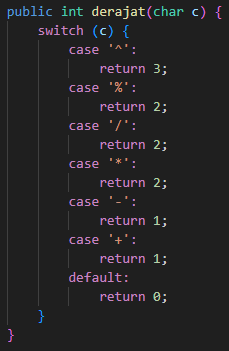
1. Buat method IsOperand dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operand.



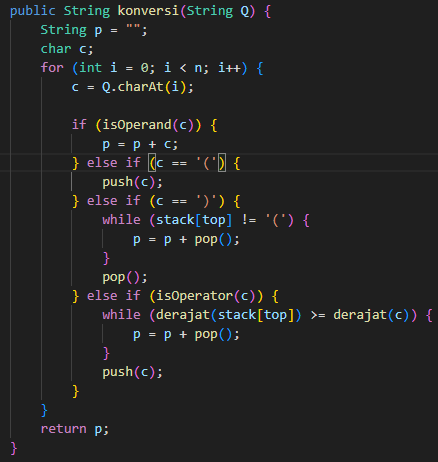
1. Buat method IsOperator dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operator.



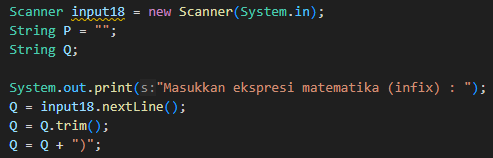
1. Buat method derajat yang mempunyai nilai kembalian integer untuk menentukan derajat operator.



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.

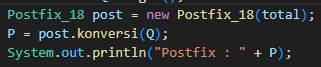


Penambahan string “)” digunakan untuk memastikan semua simbol/karakter yang masih berada di stack setelah semua persamaan terbaca, akan dikeluarkan dan dipindahkan ke postfix.

1. Buat variabel total untuk menghitung banyaknya karaketer pada variabel Q.



1. 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 serta verifikasi percobaan.



Pertanyaan Percobaan 3

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

Sebuah char akan dipindahkan atau dikeluarkan dari stack jika posisinya derajat lebih tinggi.

1. Apa fungsi kode program berikut?

c adalah inisialisaliasi type data character Q yang digunakan untuk mengambil karakter dari sebuah 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!

Karena tanda kurung hanya digunakan untuk mengatur urutan operasi, tetapi tidak diikutsertakan dalam hasil akhir postfix

**Link Repository : <https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data>**