**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**



**Oleh:**

**DZULFIKAR MUHAMMAD AL GHIFARI**

**NIM. 2341760071**

**SIB-1F / 08**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**PRAKTIKUM 7**

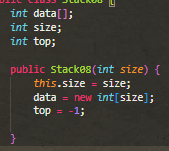
**7.2 Percobaan 1**

**6.2.1 Langkah Langkah percobaan 1**

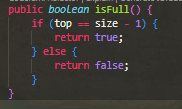
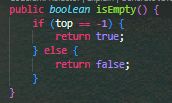
1. Membuat class sorting



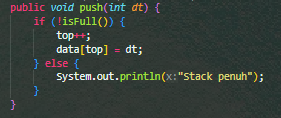
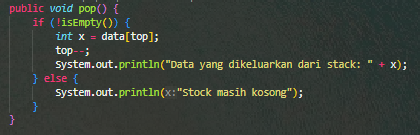
1. Deklarasi variable dan constructor



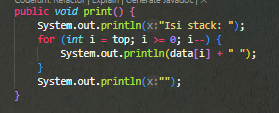
1. Menambahkan method isFull dan isEmpty

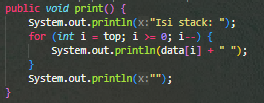
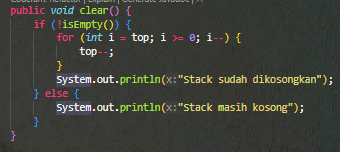
1. Menambahkan method push dan pop

1. Menambahkan method peek



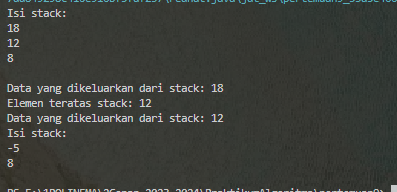
1. Menambahkan method print dan clear

1. Membuat file StackDemo dan membuat object stack



**VERIFIKASI HASIL PERCOBAAN 7.2.2**



**PERTANYAAN 3.2.3**

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.

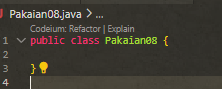
Jawaban

1. isEmpty() digunakan untuk mengecek apakah stack dalam kondisi kosong, jika stack berisi kosong maka tidak akan bisa mengambil value dan akan mengembalikan pesan stack masih kosong
2. peek() untuk memeriksa elemen paling atas, sedangkan pop() digunakan untuk mengambil elemen pada stack paling atas

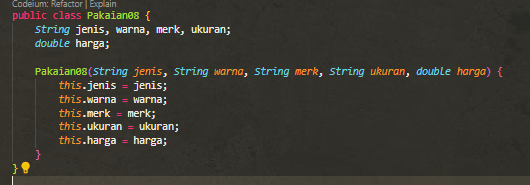
**7.3 Percobaan 2**

**7.3.1 Langkah-langkah Percobaan 2**

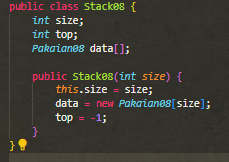
1. Menambahkan class pakaian



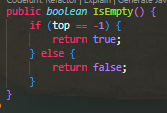
1. Menambahkan attribute dan konstruktor



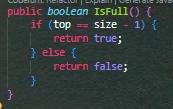
1. Menambahkan class stack dan memberi attribute dan kosntruktornya



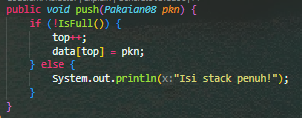
1. Menambahkan method isEmpty



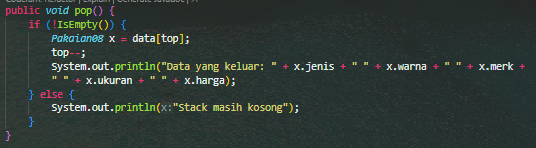
1. Menambahkan method isFull



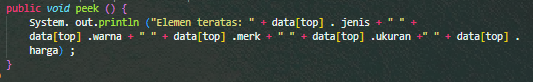
1. Menambahkan methos push



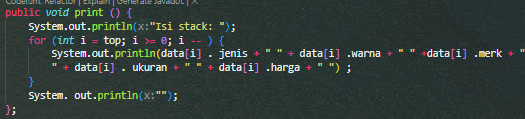
1. Menambahkan method pop



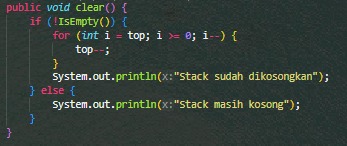
1. Menambahkan method peek



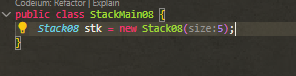
1. Membuat method print



1. Membuat method clear



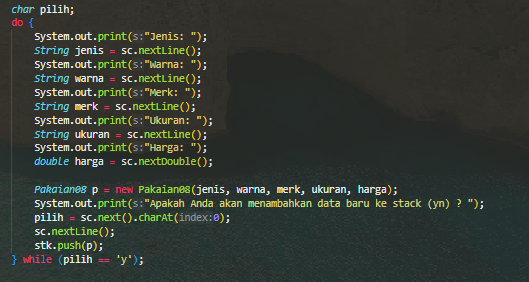
1. Membuat class stack main dan melakukan instansiasi stack



1. Deklarasi scanner



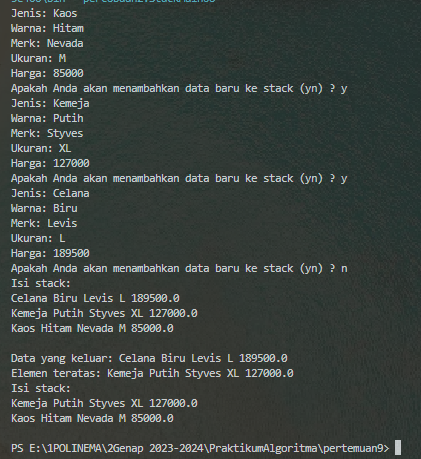
1. Menambahkan input untuk pakaian



1. Melakukan pemanggilan method



**VERIFIKASI HASIL PERCOBAAN 7.3.2**

****

**PERTANYAAN 7.3.3**

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?

1. Apakah fungsi penggunaan do-while yang terdapat pada class StackMain?
2. 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!

Jawaban

1. Ada 5 data yang bisa ditampung pada stack

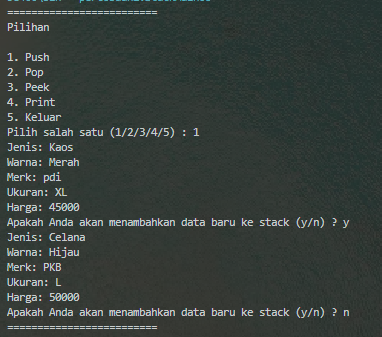


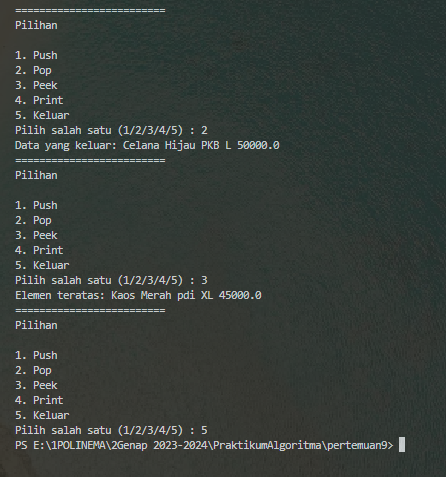
1. Data yang terkirim adalah data yang telah di inputkan pada sebelumnya



1. Untuk mengulang input pakaian sampai var pilih bernilai y
2. Dengan mengubahnya menjadi seperti berikut

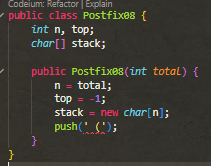


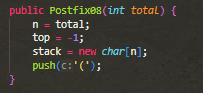


**7.4 Percobaan 3**

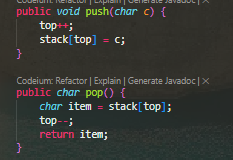
**7.4.2 Langkah-langkah Percobaan 3**

1. Membuat class postfix dan deklarasi dan membuat construct

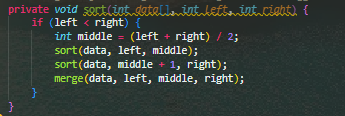




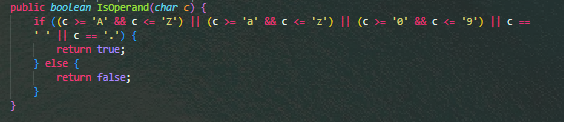
2. Menambahkan method push dan pop



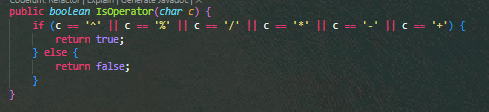
3. Menambahkan method sort



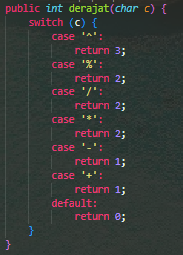
4. Membuat methos isOperand



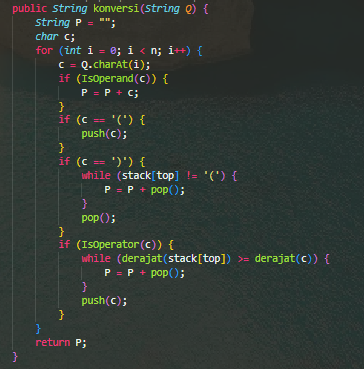
1. Membuat methos isOperator



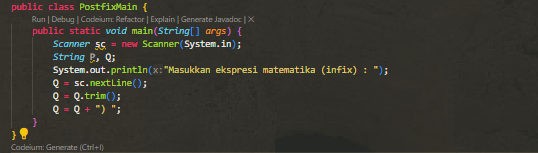
1. Membuat method derajat



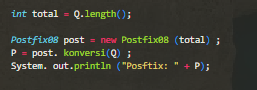
1. Membuat method konversi



1. Membuat class PostfixMain



1. Instansiasi total, post, dan pemanggilan method konversi



**VERIFIKASI HASIL PERCOBAAN 7.4.3**

****

**PERTANYAAN 7.4.3**

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 saat pemanggilan method derajar akan membawa param berupa char.

Character akan dilakukan pengecekan menggunakan switch case. Pengecekan dilakukan berurutan hingga case terakhir ( tanpa diakhiri di tengah tengah case), jika parameter sama dengan case maka akan mereturn kan angka sesuai case dan selanjutnya melakukan pengecekan pada case di bawahnya.

1. Untuk memanggil karakter urutan ke i pada Q



1. Tanda kurung tidak diperlukan pada hasil konveri, karena sudah di atur urutan nya pada konversi