**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA STRUKTUR DATA**

**STACK**



**Oleh:**

**FAUZIYYAH ADELIA RAMANDA**

**NIM. 2341760145**

**SIB-1F / 10**

**D-IV SISTEM INFORMASI BISNIS**

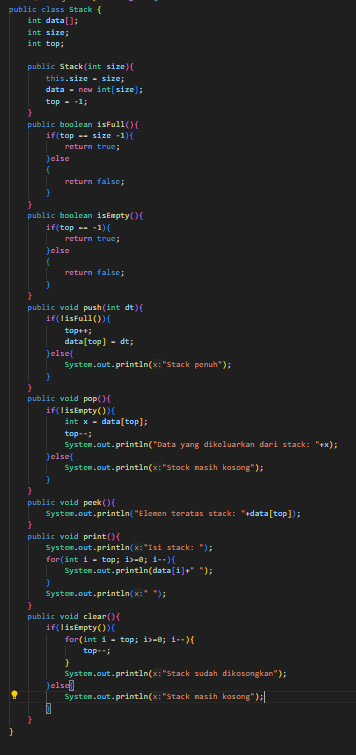
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

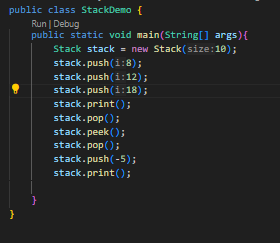
**7.2. Praktikum 1**

**7.2.1 Langkah-langkah Percobaan**

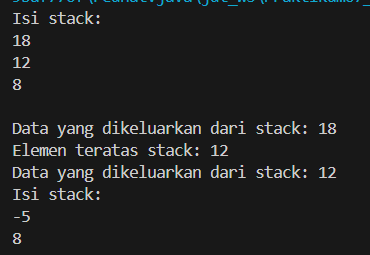
**Stack.java**

****

**StackDemo.java**

****

**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()?

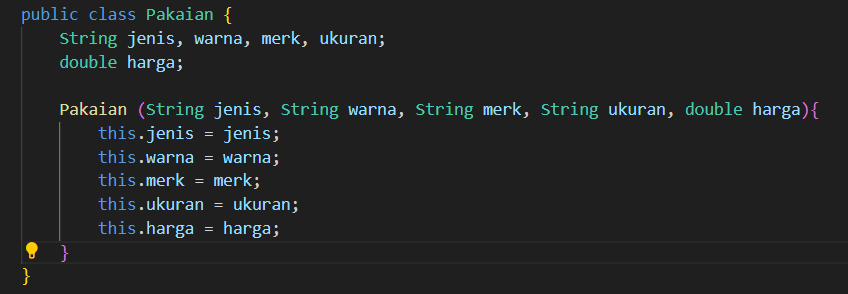
Untuk memeriksa apakah stack kosong sebelum melakukan opras pop. Jika tidak ada pemanggilan metod isEmpty(), maka program akan langsung akan melakukan operasi pengurangan indeks top tanpa mengambil data terlebih dahulu. Hal ini dapat menyebabkan eror indeks jika method pop() dipanggil saat stack kosong

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

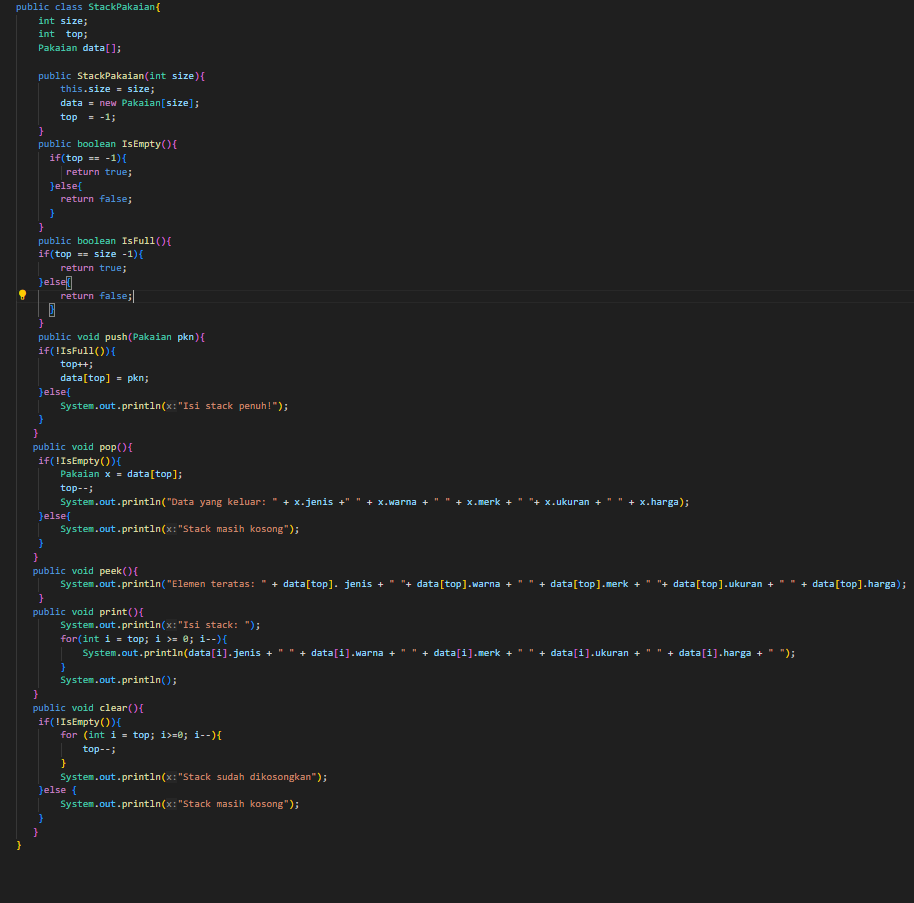
Method peek() hanya melihat elemen teratas pada stack tanpa mengubahnya, sedangkan pop() mengambil elemen paling atas dan mengurangi indeks top hingga stakc jadi lebih rendah

* 1. **Praktikum 2**
     1. **Langkah-langkah Percobaan**

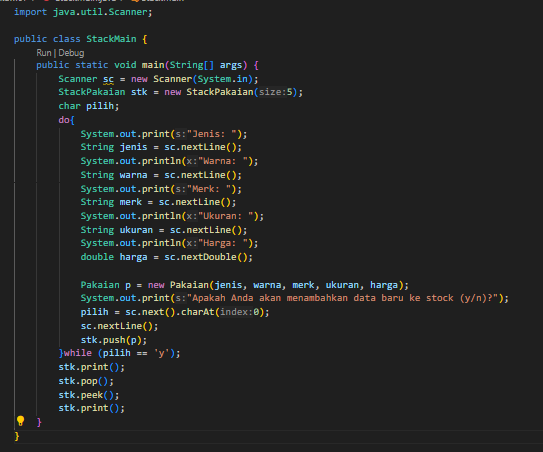
**Pakaian.java**

****

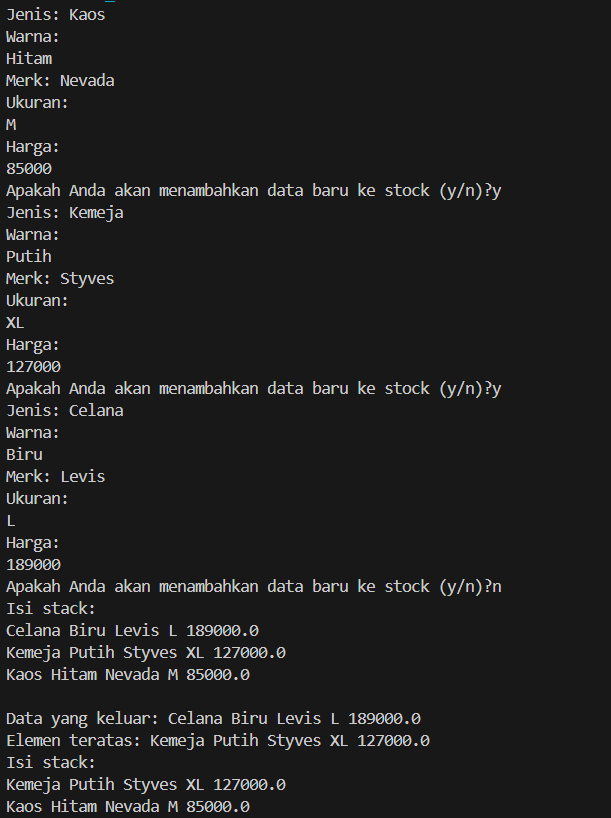
**StackPakaian.java**

****

**StackMain.java**

****

* + 1. **Verifikasi Hasil Percobaan**

****

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

5, berikut programnya



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



Data spt pakaian jenis, warna , merk, ukuran, harga

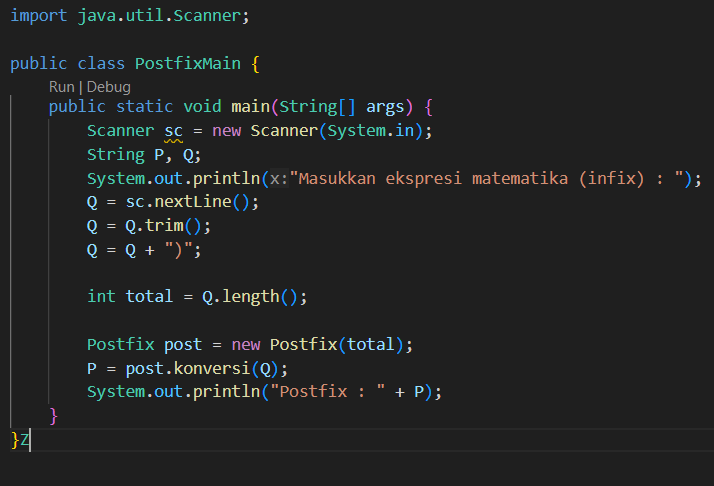
* + - 1. Apakah fungsi penggunaan **do-while** yang terdapat pada class **StackMain**?

Fungsinya untuk memasukkan secara berulang ke dalam stack hingga mencapai kapasitas maksimumnya

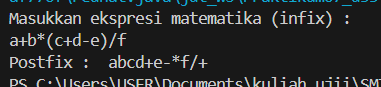
* + - 1. Modifikasi kode program pada class **StackMain** sehingga pengguna dapat memilih operasi- operasi pada stack (push, pop, peek, atau print) melalui pilihan menu program dengan memanfaatkan kondisi IF-ELSE atau SWITCH-CASE!



* 1. **Praktikum 3**
     1. **Langkah Percobaan**

****

* + 1. **Verifikasi Hasil Percobaan**



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

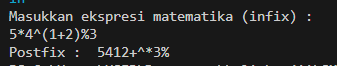
Method derajat menghitung derajat dari operator yang diinput, tapii a digunaka dalam method konversi untuk mengkonversi infix expression menjadi postfix expression.

* + - 1. Apa fungsi kode program berikut?



Fungsinya untuk mengambil karakter pada posisi I dan string Q, dan menyimpannya dalam variabel c untuk diproses selanjutnya

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



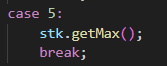
* + - 1. Pada soal nomor 3, mengapa tanda kurung tidak ditampilkan pada hasil konversi? Jelaskan!

Karena tanda kurung bukan drajat operator aritmatika

**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!





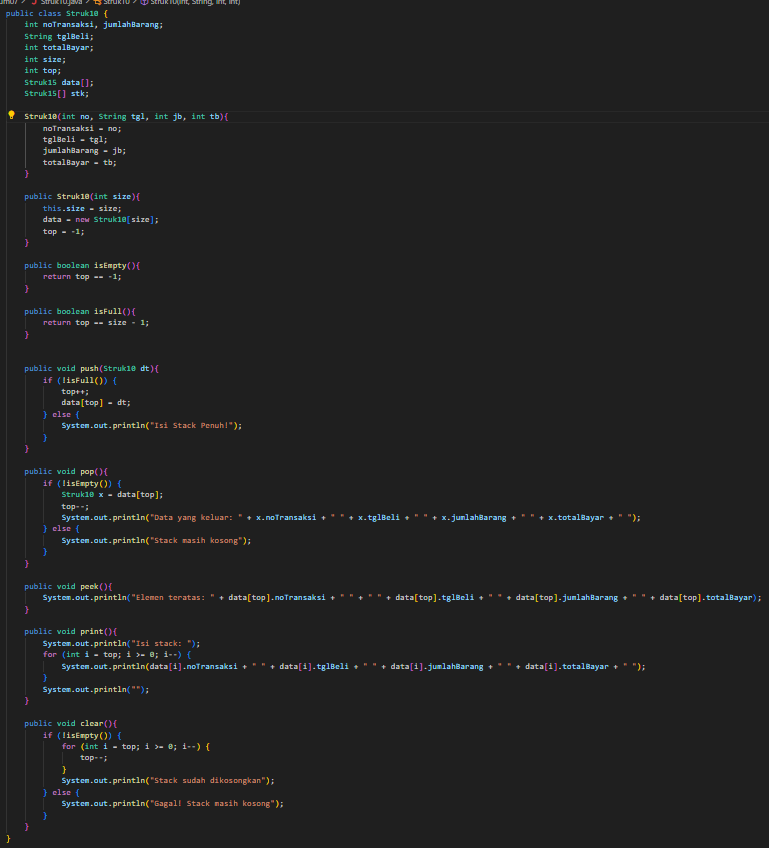
1. Setiap hari Minggu, Dewi pergi berbelanja ke salah satu supermarket yang berada di area rumahnya. Setiap kali selesai berbelanja, Dewi menyimpan struk belanjaannya di dalam laci. Setelah dua bulan, ternyata Dewi sudah mempunyai delapan struk belanja. Dewi berencana mengambil lima struk belanja untuk 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:

* + Nomor transaksi
  + Tanggal pembelian
  + Jumlah barang yang dibeli
  + Total harga bayar

Tampilkan informasi struk belanja yang masih tersimpan di dalam stack!

**Struk10.java**

****

**StrukMain10.java**

****