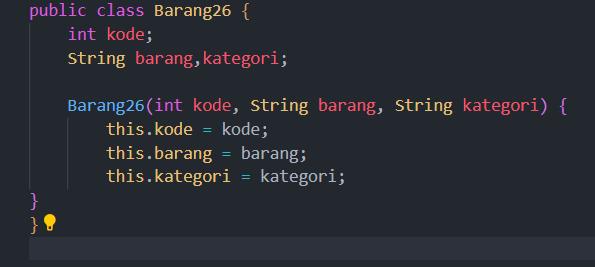
**Jobsheet 8 Stacking**

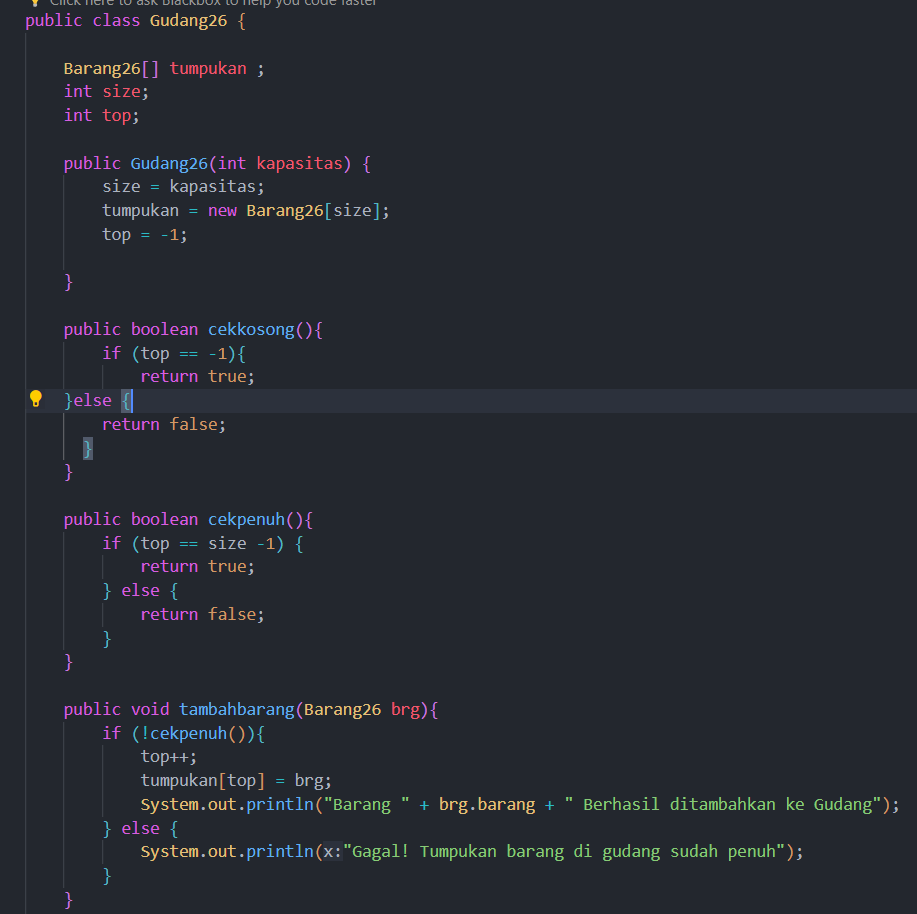
**Satria Wiguna/ Ti 1d / Absen 26**

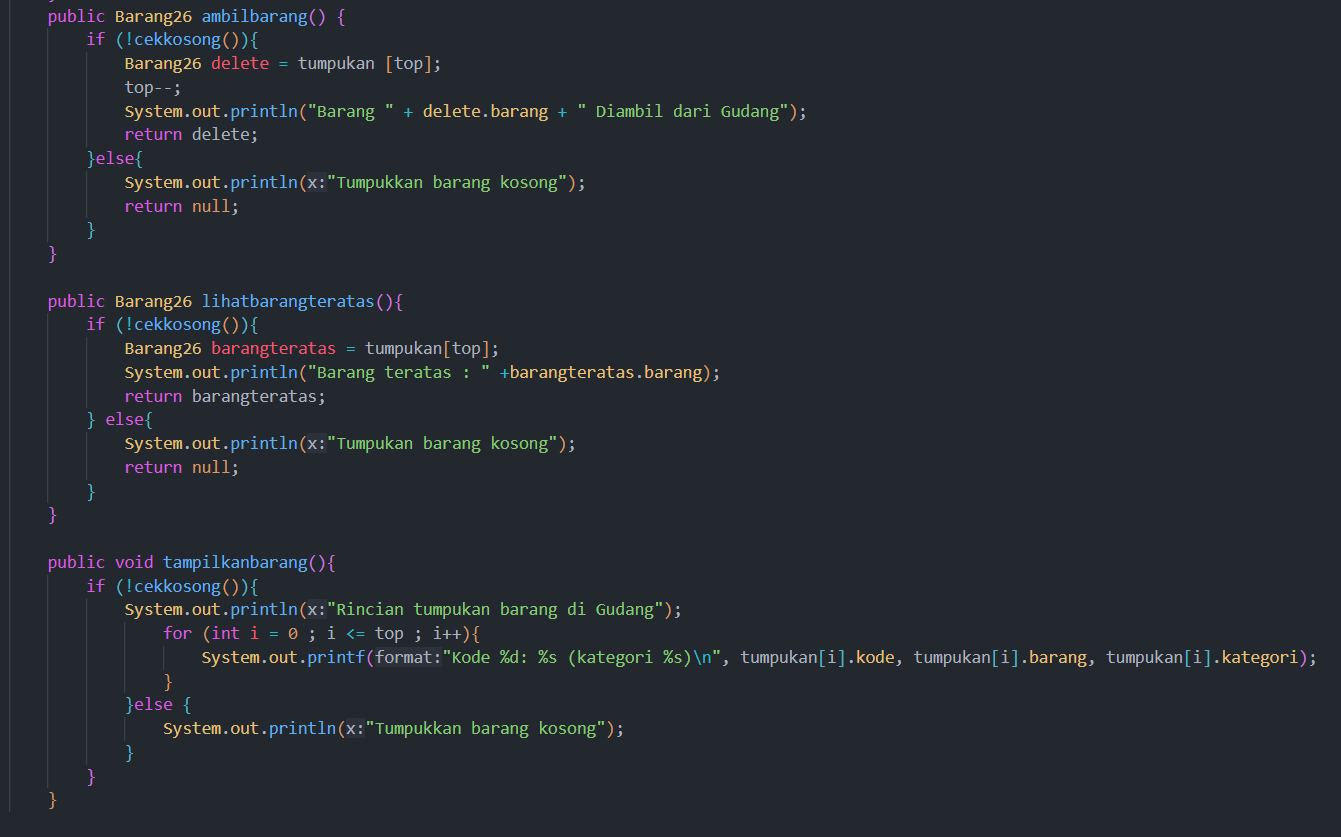
**Percobaan 1 :**

**Barang26:**

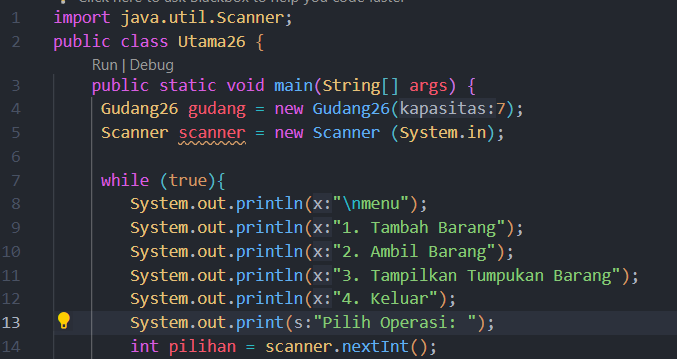
****

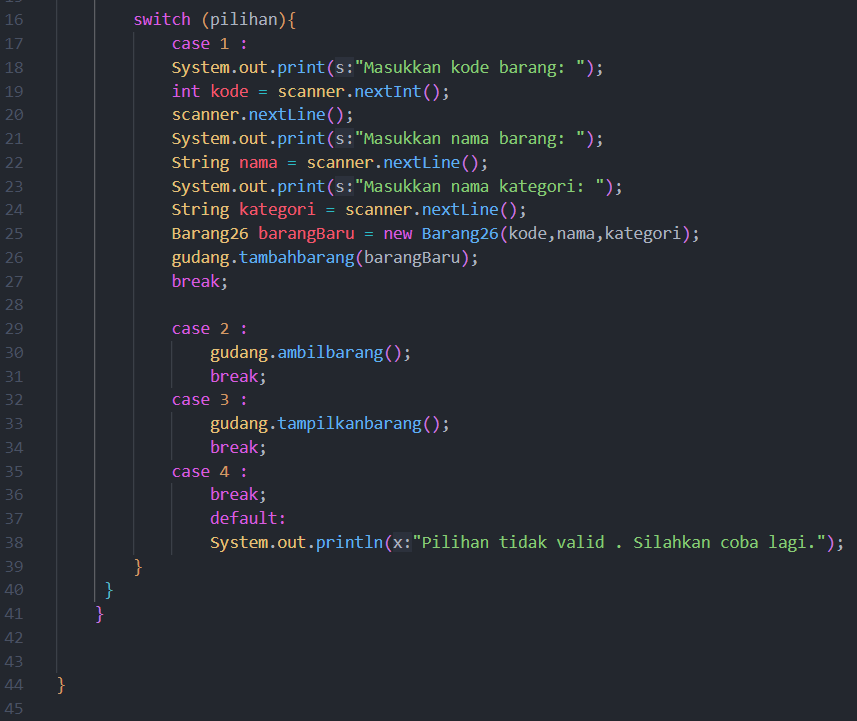
**Gudang26:**

****

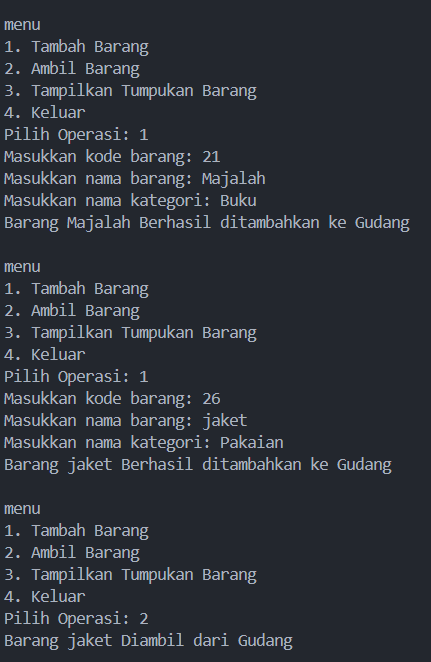
****

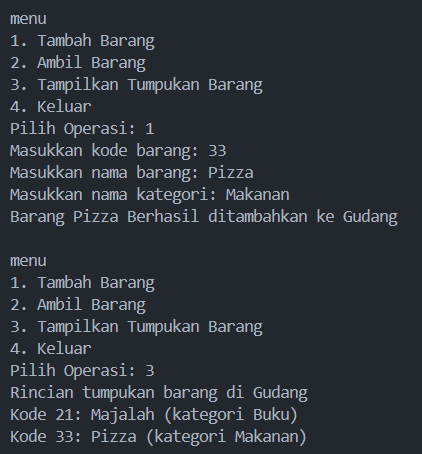
**Utama26:**

****

****

**Vervikasi hasil :**

****

****

2.1.3 Pertanyaan

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana saja yang perlu diperbaiki?

perlu diperbaiki pada bagian metode tampilkanBarang, terdapat kesalahan dalam format string pada printf. Format %f digunakan untuk menampilkan tipe data float, sedangkan nama pada Barang10 bertipe data string.

1. Berapa banyak data barang yang dapat ditampung di dalam tumpukan? Tunjukkan potongan kode programnya!



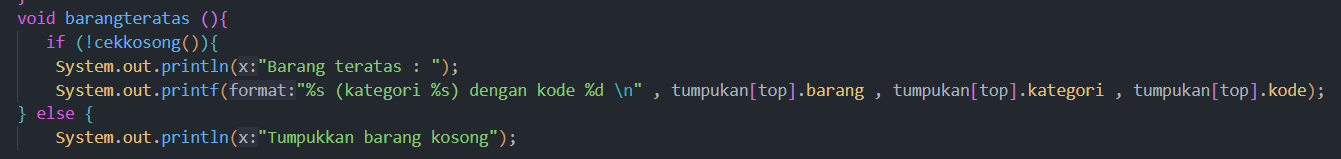
1. Mengapa perlu pengecekan kondisi !cekKosong() pada method tampilkanBarang? Kalau kondisi tersebut dihapus, apa dampaknya?

untuk memastikan bahwa tumpukan tidak kosong sebelum mencoba menampilkan barang. Jika kondisi tersebut dihapus, maka program akan mencoba menampilkan barang bahkan ketika tumpukan kosong.

1. Modifikasi kode program pada class Utama sehingga pengguna juga dapat memilih operasi lihat barang teratas, serta dapat secara bebas menentukan kapasitas gudang!

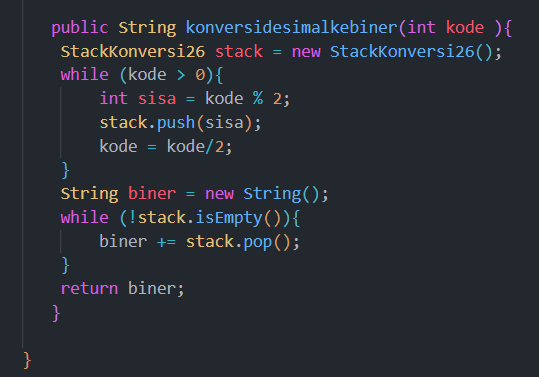
Utama26:

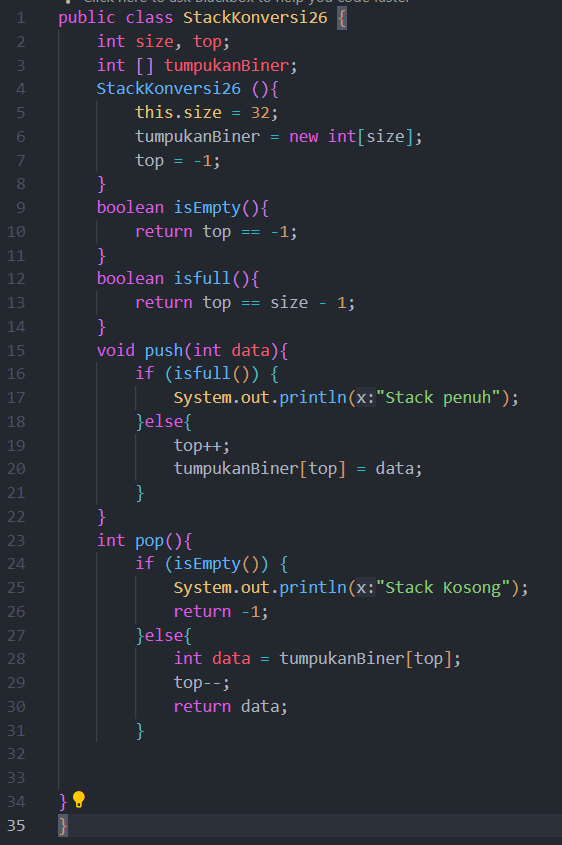


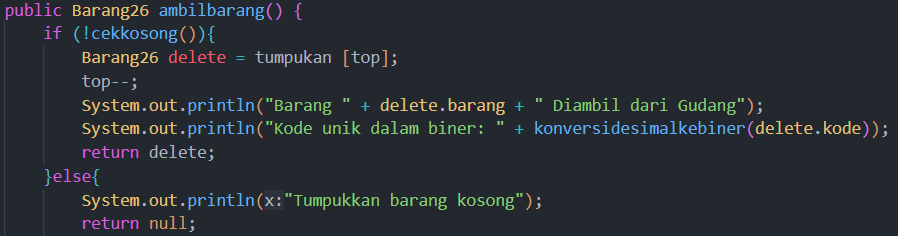
Gudang26:

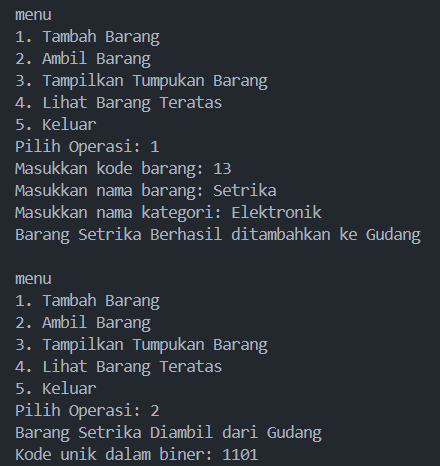
1. Commit dan push kode program ke Github

**Percobaan 2**

****

****

****

****

**Pertanyaan**1. Pada method **konversiDesimalKeBiner**, ubah kondisi perulangan menjadi **while (kode != 0)**,  
bagaimana hasilnya? Jelaskan alasannya!

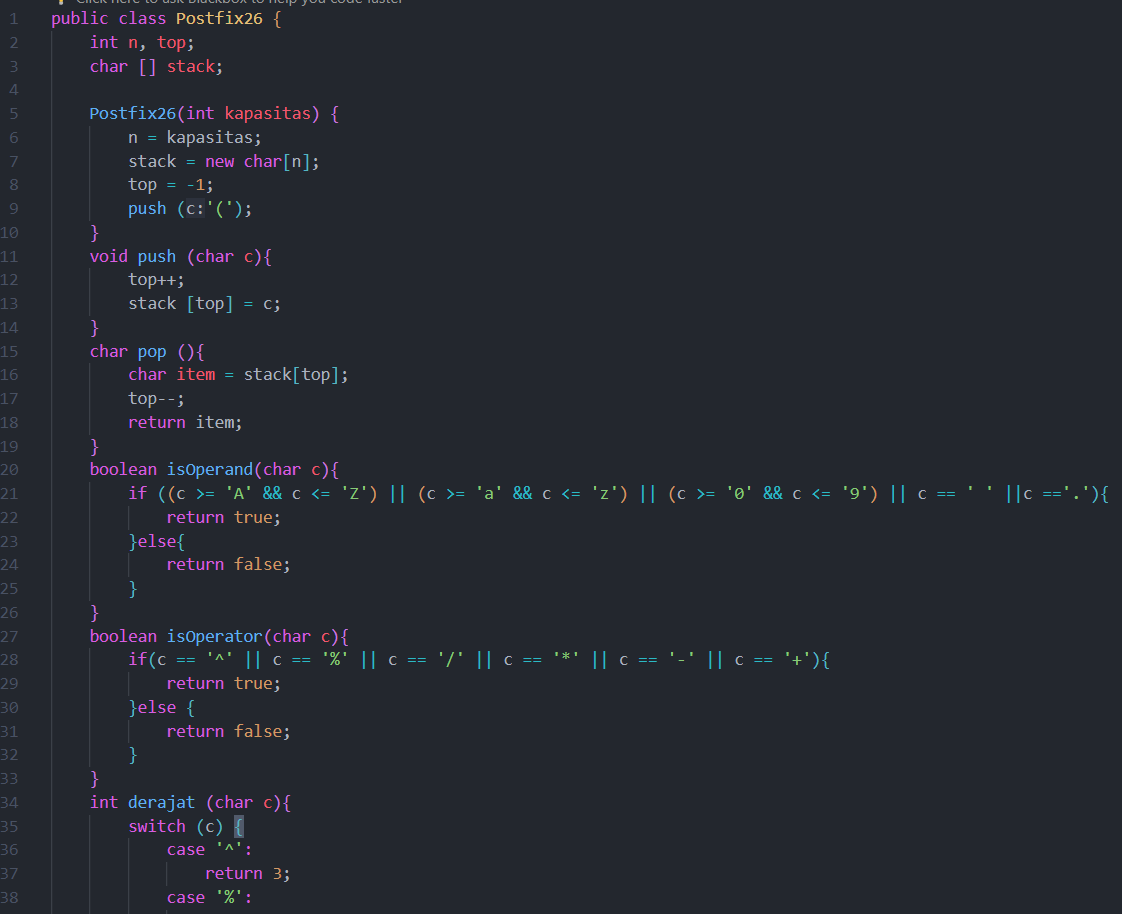
Setelah diubah menjadi while (kode != 0), perulangan akan dilakukan selama kode tidak sama dengan 0. dan hasilnya pun sama ketika menggunakan while (kode != 0)

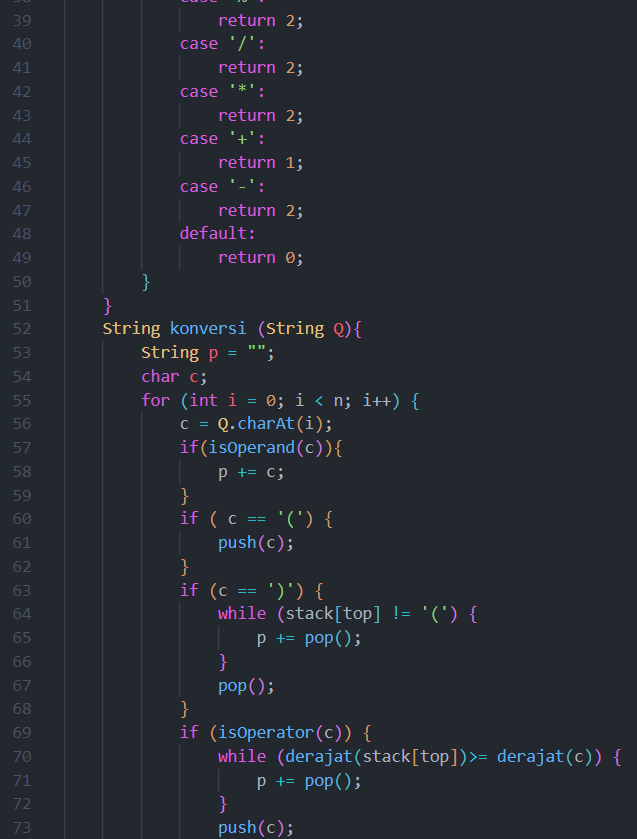
2. Jelaskan alur kerja dari method **konversiDesimalKeBiner**!

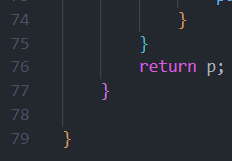
* objek StackKonversi10 dibuat untuk menyimpan sisa pembagian bilangan desimal dengan 2, Selama kode tidak sama dengan 0 (while (kode != 0)):
* kemudian menghitung sisa pembagian kode dengan 2 (int sisa = kode % 2).
* lalu memasukkan sisa tersebut ke dalam stack menggunakan metode push. setelah itu membagi kode dengan 2 untuk memperoleh nilai kode baru yang akan digunakan pada iterasi berikutnya.
* Setelah semua sisa pembagian dimasukkan ke dalam stack, lakukan iterasi melalui stack untuk mengambil nilai sisa dan tambahkan ke string biner menggunakan StringBuilder.
* lalu mengembalikan string biner yang berisi representasi biner dari bilangan desimal.

**Percobaan 3 :**

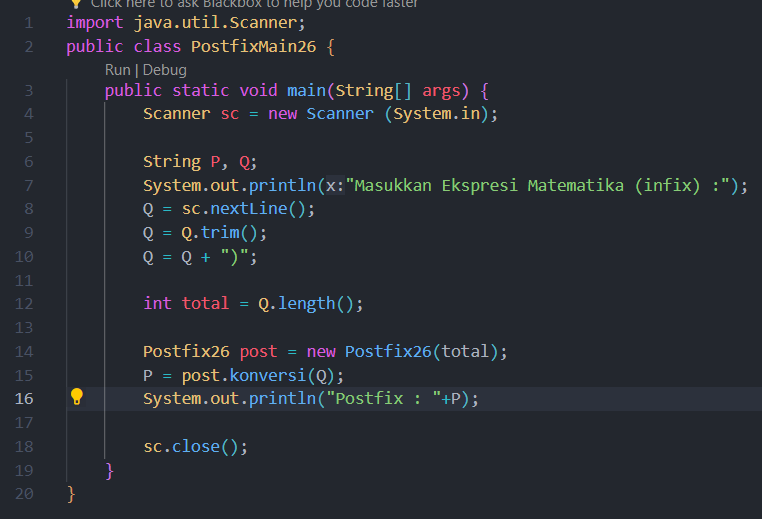
Postfix26 :







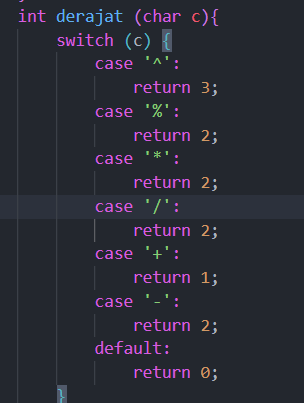
PostfixMain26 :

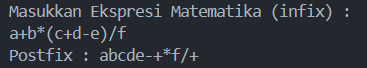


**Pertanyaan**1. Pada method **derajat**, mengapa return value beberapa case bernilai sama? Apabila return  
value diubah dengan nilai berbeda-beda setiap case-nya, apa yang terjadi?

Operator dengan prioritas lebih tinggi harus dievaluasi terlebih dahulu. Jika mengubah return value menjadi angka yang berbeda untuk setiap kasus, urutan prioritas dalam mengevaluasi operator akan terganggu

Contoh kita mengubah menjadi





Hasil ahan terlihat menjadi  
2. Jelaskan alur kerja method **konversi**!

* Method konversi mengubah ekspresi infix menjadi ekspresi postfix menggunakan stack, serta menginisialisasi string kosong p untuk menyimpan ekspresi postfix.
* Method melakukan looping untuk setiap karakter dalam ekspresi infix Q.
* Jika karakter adalah operand (huruf, digit, spasi, atau titik), karakter tersebut langsung ditambahkan ke string postfix p.
* Jika karakter adalah tanda kurung buka (, ia dipush ke stack.
* Jika karakter adalah tanda kurung tutup ), operator-oprator dipop dari stack dan ditambahkan ke p sampai menemukan tanda kurung buka.
* Jika karakter adalah operator (^, %, /, \*, -, +), ia membandingkan prioritasnya dengan operator teratas di stack.
* Jika prioritas operator saat ini lebih rendah atau sama dengan operator teratas di stack, operator teratas dipop dan ditambahkan ke p. Proses ini terus berlanjut sampai stack kosong atau operator teratas memiliki prioritas yang lebih rendah. Kemudian, operator saat ini dipush ke stack.
* Setelah semua karakter dalam Q diproses, operator-operator yang tersisa di stack dipop dan ditambahkan ke p untuk melengkapi ekspresi postfix.
* Method mereturn value postfix p.

3. Pada method konversi, apa fungsi dari potongan kode berikut?



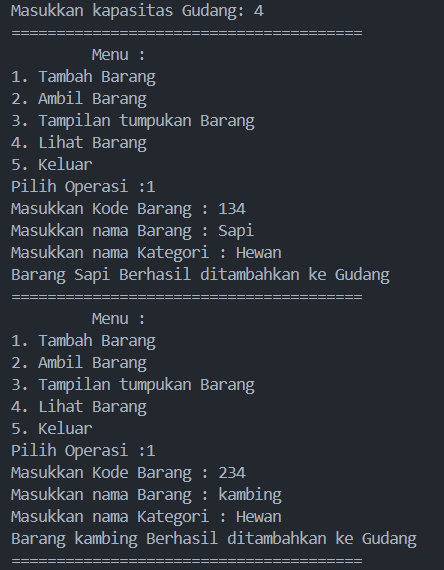
kode Q.length() berfungsi untuk mendapatkan panjang dari string Q, Panjang string ini digunakan untuk menginisialisasi variabel total, yang nantinya digunakan untuk mengembalikan jumlah karakter dalam string Q,

**Latihan Praktikum  
Waktu : 60 Menit**Perhatikan dan gunakan kembali kode program pada Percobaan 1. Tambahkan dua method berikut  
pada class Gudang:  
• Method **lihatBarangTerbawah** digunakan untuk mengecek barang pada tumpukan terbawah

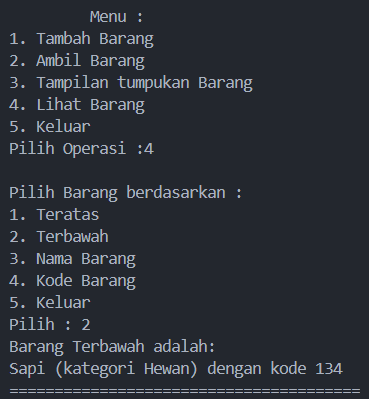


• Method **cariBarang** digunakan untuk mencari ada atau tidaknya barang berdasarkan kode  
barangnya atau nama barangnya

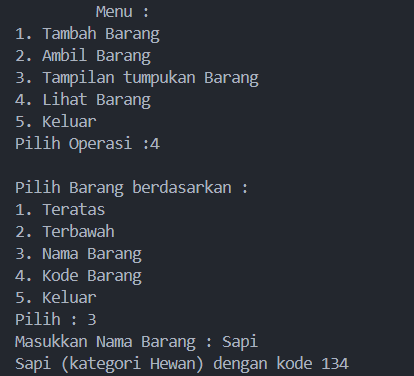




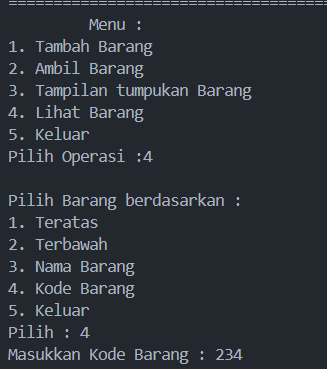
Berdasarkan terbawah



Berdasarkan nama



Berdasarkan kode :



Link github :

https://github.com/AuroraSauces/Praktikum-algoritma-dan-sistem-data/tree/main