**LAPORAN PRAKTIKUM**

**Politeknik Negeri Malang**

**Praktikan**

**ALGORITMA DAN STRUKTUR DATA**

****

2022

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi [harus diupdate setiap ada perubahan]

[Praktikum 1 4](#_Toc100737972)

[Langkah 1 4](#_Toc100737973)

[Langkah 2 4](#_Toc100737974)

[Langkah 3 4](#_Toc100737975)

[Langkah 4 5](#_Toc100737976)

[Langkah 5 5](#_Toc100737977)

[Langkah 6 6](#_Toc100737978)

[Langkah 7 6](#_Toc100737979)

[Langkah 8 7](#_Toc100737980)

[Langkah 9 7](#_Toc100737981)

[Langkah 10 7](#_Toc100737982)

[Langkah 11 7](#_Toc100737983)

[Langkah 12 7](#_Toc100737984)

[Langkah 13 8](#_Toc100737985)

[Langkah 14 8](#_Toc100737986)

[Langkah 15 8](#_Toc100737987)

[Langkah 16 9](#_Toc100737988)

[Langkah 17 9](#_Toc100737989)

[Verifikasi Hasil Percobaan 9](#_Toc100737990)

[Pertanyaan 9](#_Toc100737991)

[Jawaban 10](#_Toc100737992)

[Praktikum 2 12](#_Toc100737993)

[Langkah 1 12](#_Toc100737994)

[Langkah 2 12](#_Toc100737995)

[Langkah 3 12](#_Toc100737996)

[Langkah 4 13](#_Toc100737997)

[Langkah 5 13](#_Toc100737998)

[Langkah 6 14](#_Toc100737999)

[Langkah 7 14](#_Toc100738000)

[Langkah 8 15](#_Toc100738001)

[Langkah 9 15](#_Toc100738002)

[Langkah 10 16](#_Toc100738003)

[Langkah 11 16](#_Toc100738004)

[Langkah 12 16](#_Toc100738005)

[Verifikasi Hasil Percobaan 16](#_Toc100738006)

[Pertanyaan 17](#_Toc100738007)

[Jawaban 17](#_Toc100738008)

[Tugas 18](#_Toc100738009)

[Soal Nomor 1 18](#_Toc100738010)

[Soal Nomor 2 18](#_Toc100738011)

[Jawaban Nomor 1 18](#_Toc100738012)

[Source code yang dimodifikasi pada class StackRidwan 18](#_Toc100738013)

[Source code yang dimodifikasi pada class StackMainRidwan 19](#_Toc100738014)

[Output 20](#_Toc100738015)

[Jawaban Nomor 2 21](#_Toc100738016)

[Source Code StrukBelanjaRidwan 21](#_Toc100738017)

[Source Code StackStrukRidwan 21](#_Toc100738018)

[Source Code StrukMain 23](#_Toc100738019)

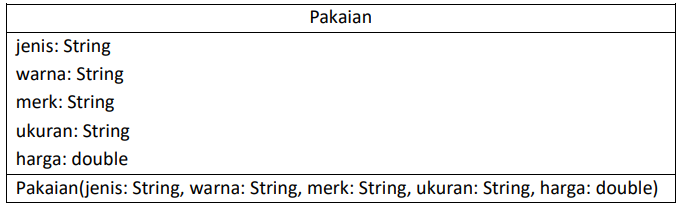
[Output 24](#_Toc100738020)

# Praktikum 1

Pada percobaan ini, kita akan membuat program yang mengilustrasikan tumpukan pakaian yang disimpan ke dalam stack. Karena sebuah pakaian mempunyai beberapa informasi, maka implementasi Stack dilakukan dengan menggunakan array of object untuk mewakili setiap elemennya.

## Langkah 1

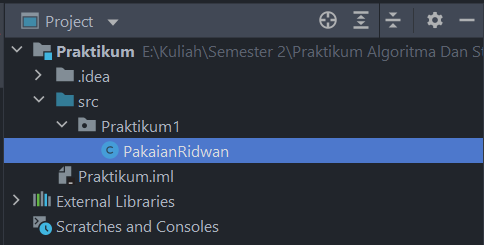
Perhatikan Diagram Class Pakaian berikut ini:



Berdasarkan diagram class tersebut, akan dibuat program class Pakaian dalam Jawa

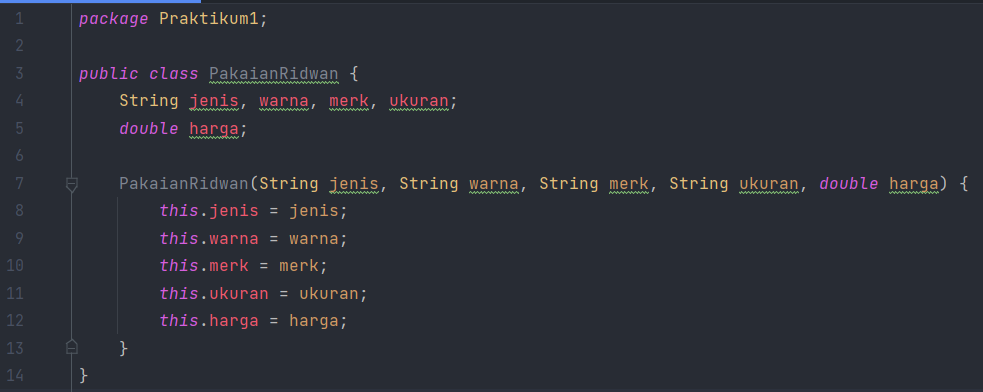
## Langkah 2

Buat package dengan nama Praktikum1, kemudian buat class baru dengan nama Pakaian.



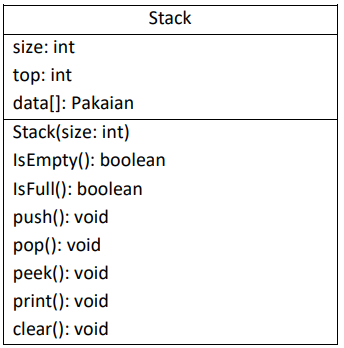
## Langkah 3

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



## Langkah 4

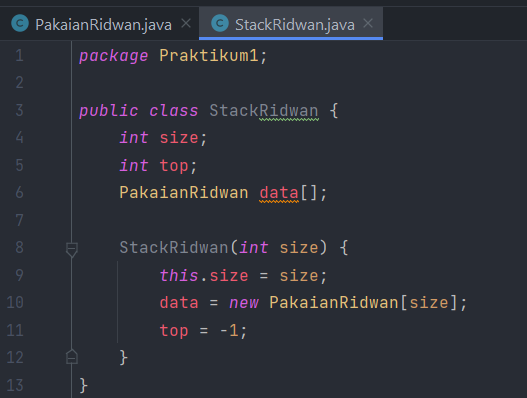
Setelah membuat class Pakaian, selanjutnya perlu dibuat class Stack yang berisi atribut dan method sesuai diagram Class Stack berikut ini:



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.

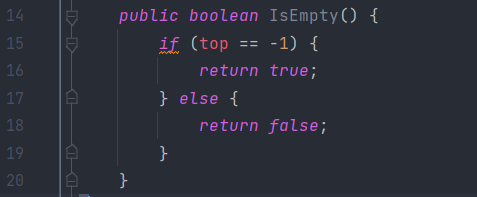
## Langkah 5

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



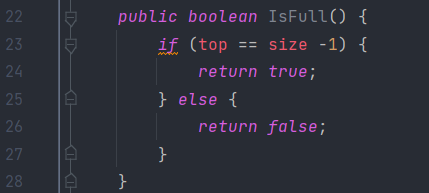
## Langkah 6

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



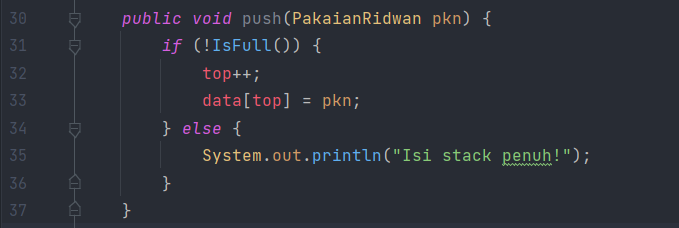
## Langkah 7

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



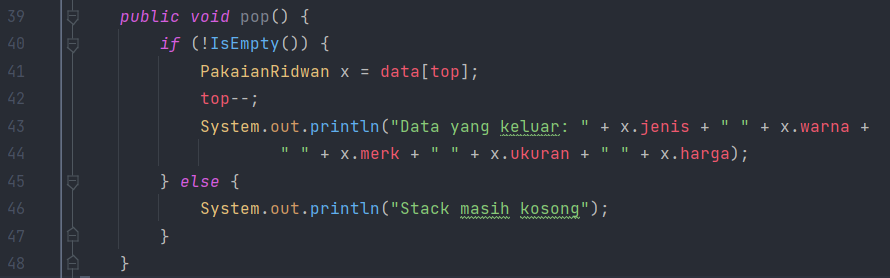
## Langkah 8

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



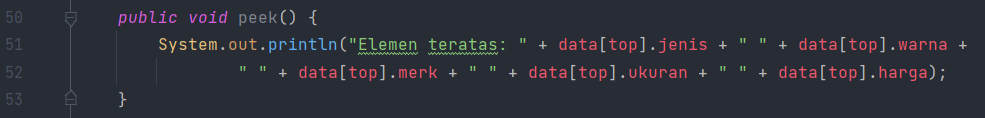
## Langkah 9

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



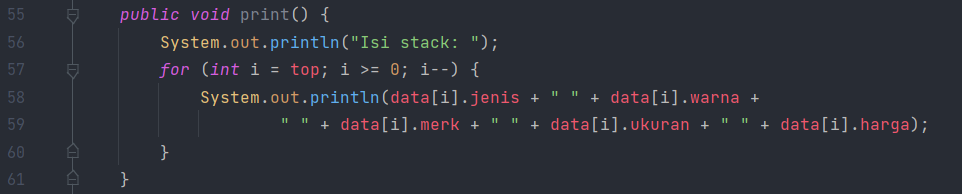
## Langkah 10

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



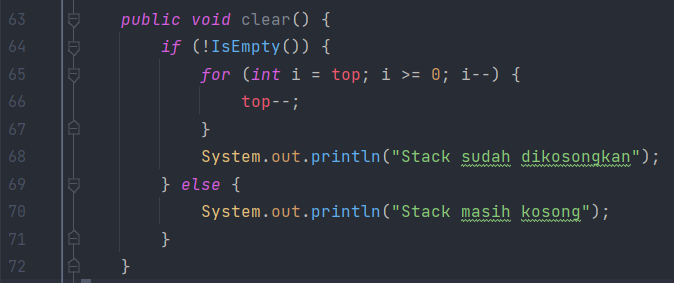
## Langkah 11

Buat method print bertipe void untuk menampilkan seluruh elemen pada stack



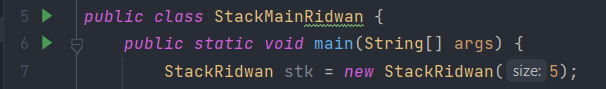
## Langkah 12

Buat method clear bertipe void untuk menghapus seluruh isi stack.



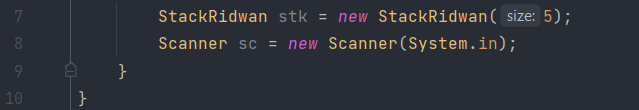
## Langkah 13

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.



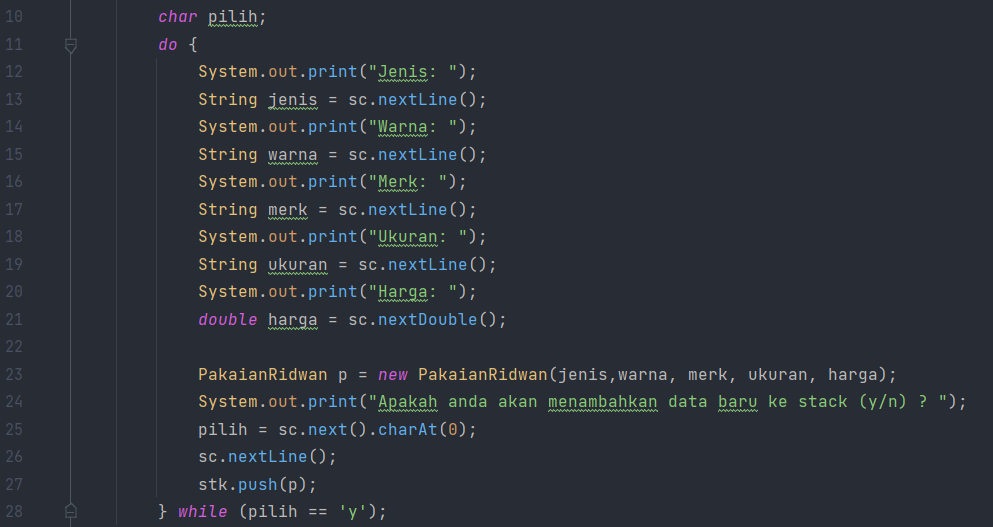
## Langkah 14

Deklarasikan Scanner dengan nama sc



## Langkah 15

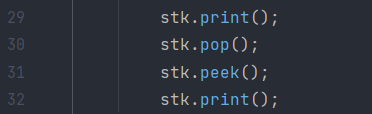
Tambahkan kode berikut ini untuk menerima input data Pakaian, kemudian semua informasi tersebut dimasukkan ke dalam stack



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

## Langkah 16

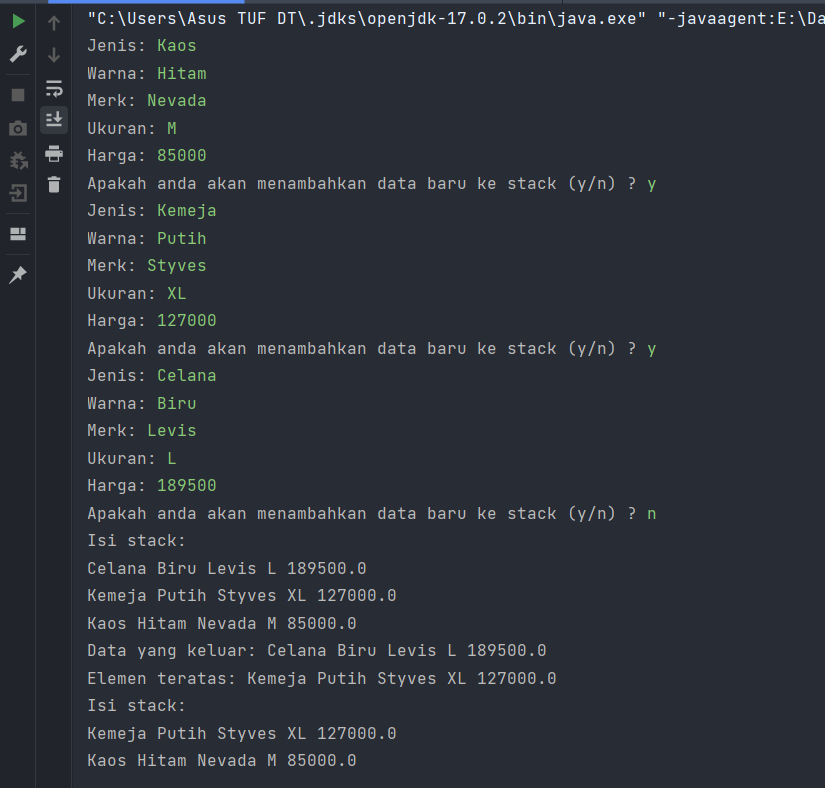
Lakukan pemanggilan method print, method pop, dan method peek dengan urutan sebagai berikut.



## Langkah 17

Compile dan jalankan class StackMain, kemudian amati hasilnya.

## Verifikasi Hasil Percobaan



### 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?

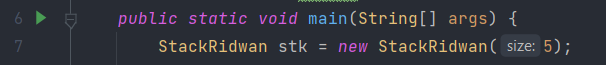


1. Apakah fungsi penggunaan do-while yang terdapat pada class StackMain?
2. Modifikasi kode program pada class StackMain sehingga pengguna dapat memilih operasioperasi pada stack (push, pop, peek, atau print) melalui pilihan menu program dengan memanfaatkan kondisi IF-ELSE atau SWITCH-CASE!

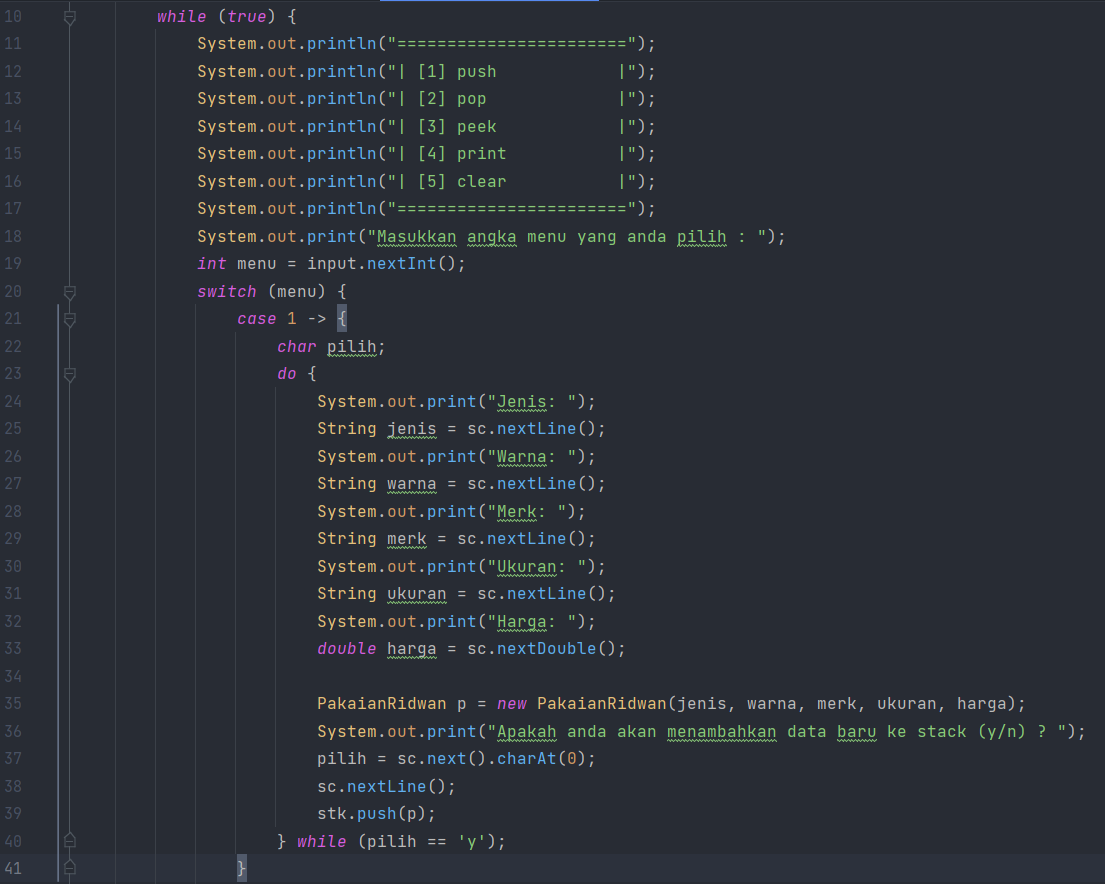
### Jawaban

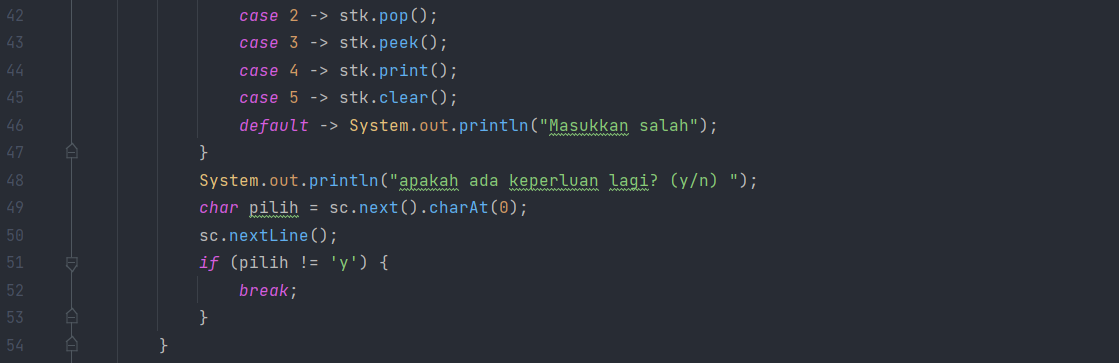
1. Banyak data pakaian yang disimpan pada stack ada 5.

Pada kode dibawah ini memanggil konstruktor StackRidwan dengan parameter bernilai 5.

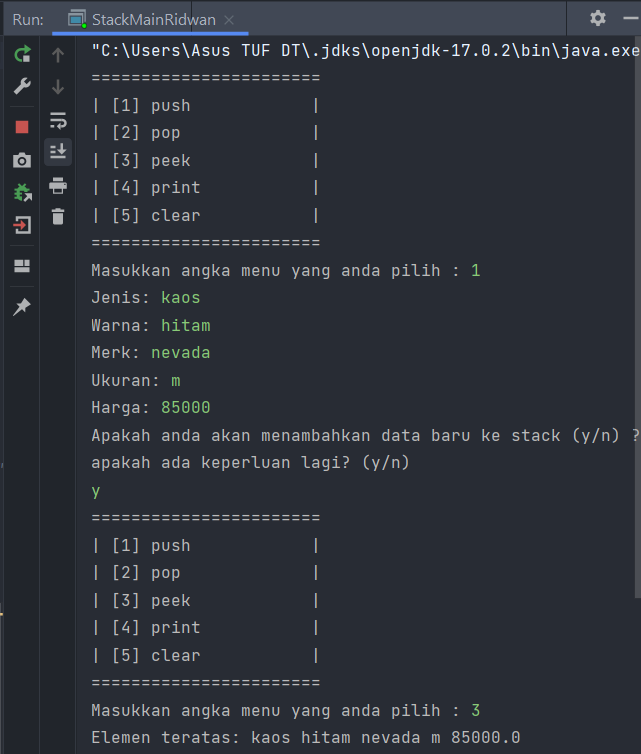


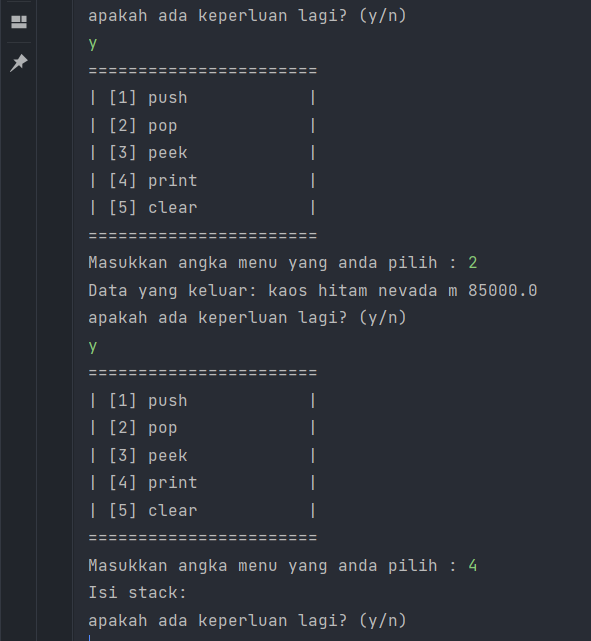
1. Pada variabel data p disimpan data dengan tipe data PakaianRidwan yang memiliki attribute Jenis, Warna, Merk, Ukuran, Harga.
2. Penggunaan do while agar perulangan setidaknya dilakukan minimal satu kali.
3. Source code class StackMainRidwan





Output



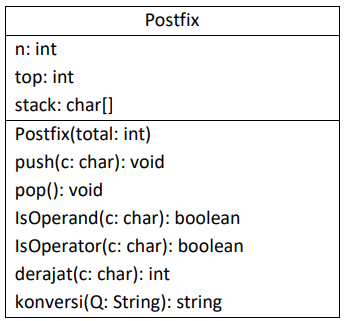


# Praktikum 2

Pada percobaan ini, kita akan membuat program untuk melakukan konversi notasi infix menjadi notasi postfix.

## Langkah 1

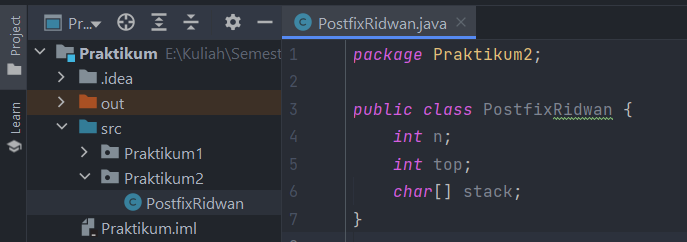
Perhatikan Diagram Class berikut ini:



Berdasarkan diagram class tersebut, akan dibuat program class Postfix dalam Java.

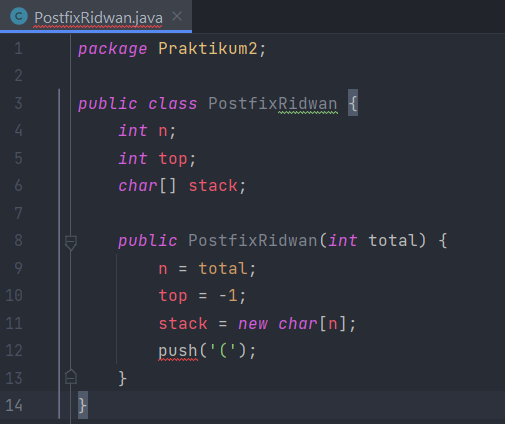
## Langkah 2

Buat package dengan nama Praktikum2, kemudian buat class baru dengan nama Postfix. Tambahkan atribut n, top, dan stack sesuai diagram class Postfix tersebut.



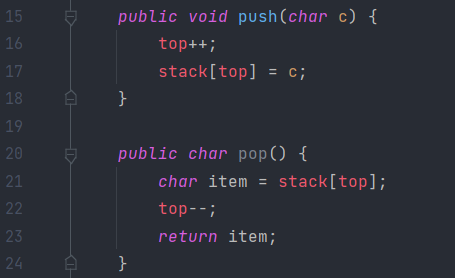
## Langkah 3

Tambahkan pula konstruktor berparameter seperti gambar berikut ini.



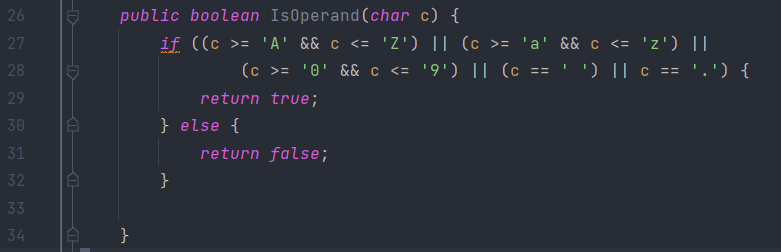
## Langkah 4

Buat method push dan pop bertipe void



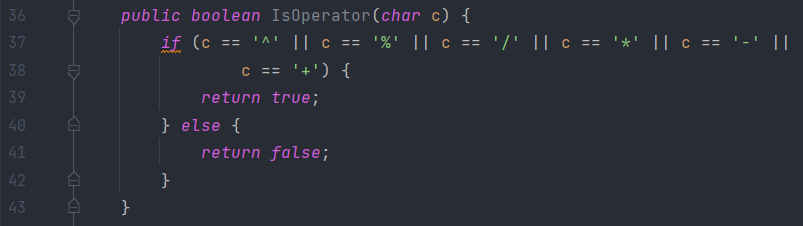
## Langkah 5

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



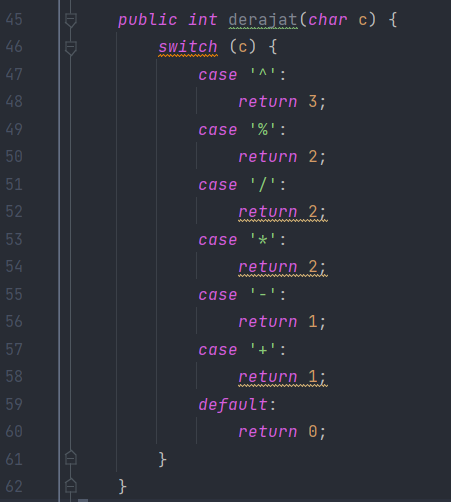
## Langkah 6

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



## Langkah 7

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



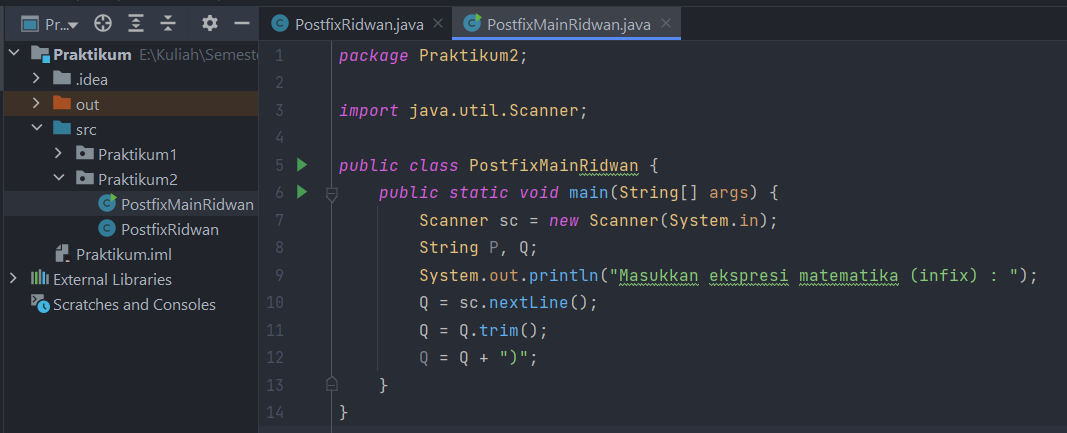
## Langkah 8

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



## Langkah 9

Selanjutnya, buat class baru dengan nama PostfixMain tetap pada package Praktikum2. 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.

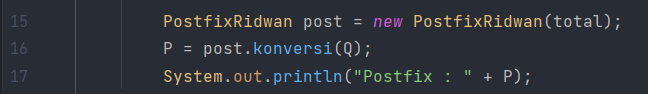
## Langkah 10

Buat variabel total untuk menghitung banyaknya karaketer pada variabel Q.



## Langkah 11

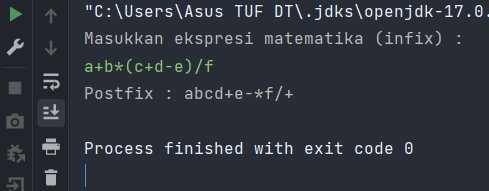
Lakukan instansiasi objek dengan nama post dan nilai parameternya adalah total. Kemudian panggil method konversi untuk melakukan konversi notasi infix Q menjadi notasi postfix P.



## Langkah 12

Compile dan jalankan class PostfixMain dan amati hasilnya

## Verifikasi Hasil Percobaan



### Pertanyaan

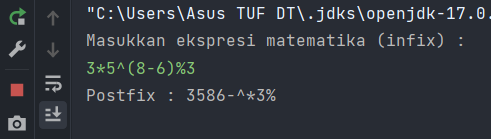
1. Perhatikan class Postfix, jelaskan alur kerja method derajat!
2. Apa fungsi kode program berikut?



1. Jalankan kembali program tersebut, masukkan ekspresi 3\*5^(8-6)%3. Tampilkan hasilnya!
2. Pada soal nomor 3, mengapa tanda kurung tidak ditampilkan pada hasil konversi? Jelaskan!

### Jawaban

1. Pada method derajat memiliki parameter c yaitu tipedata char kemudian melakukan switch c. dari nilai c tersebut ditentukan jika nilai c adalah tanda ‘^’ maka mengembalikan nilai derajat 3. Jika nilai c adalah tanda ‘%’ atau ‘/’ atau ‘\*’ maka mengembalikan nilai derajat 2. Dan jika nilai c adalah tanda ‘-‘ atau ‘+’ maka mengembalikan nilai 1, dan memiliki nilai kembalian default 0.
2. Kode program tersebut berfungsi untuk mengambil character dari string Q dengan indeks sesuai dengan iterasi i.
3. Output



1. Karena jika nilai c = ‘(‘ maka akan dilakukan push ke stack saja, dan jika nilai c = ‘)’ maka akan dilakukan pop dan memasukkan nilai yang di pop ke String P proses ini diulang hingga menemukan tanda ‘(‘ dan tanda ‘(’ dihapus.

# Tugas

## Soal Nomor 1

Perhatikan dan gunakan kembali kode program pada Praktikum 1. Tambahkan method getMin pada class Stack yang digunakan untuk mencari dan menampilkan data pakaian dengan harga terendah dari semua data pakaian yang tersimpan di dalam stack!

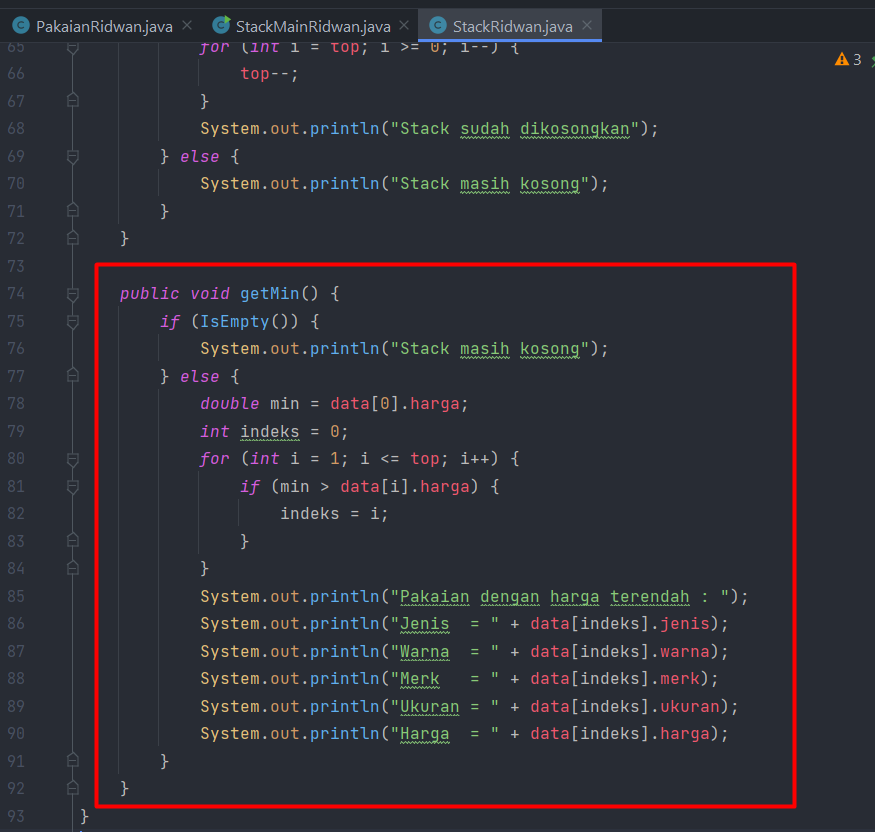
## Soal Nomor 2

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:

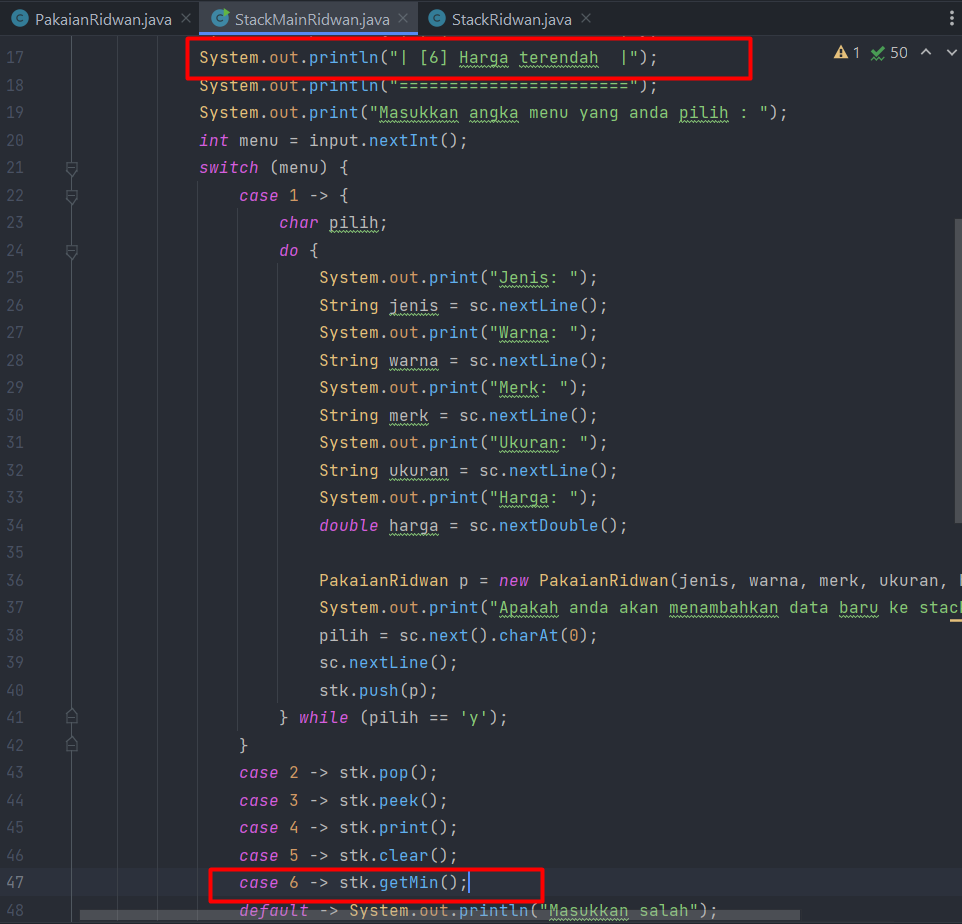
* Nomor transaksi
* Tanggal pembelian
* Jumlah barang yang dibeli
* Total harga bayar

## Jawaban Nomor 1

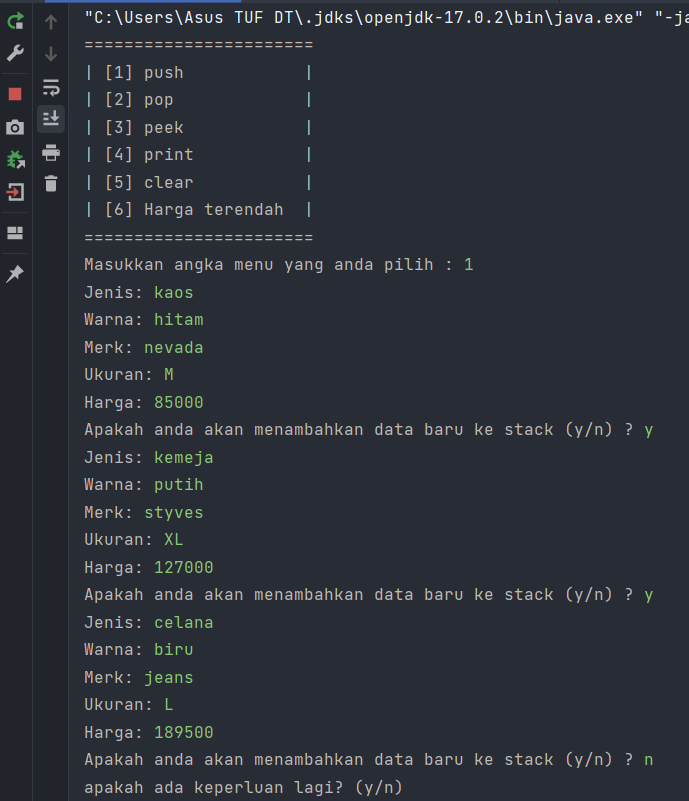
### Source code yang dimodifikasi pada class StackRidwan

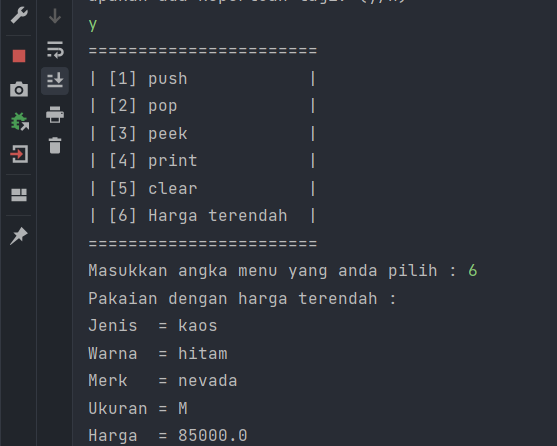


### Source code yang dimodifikasi pada class StackMainRidwan



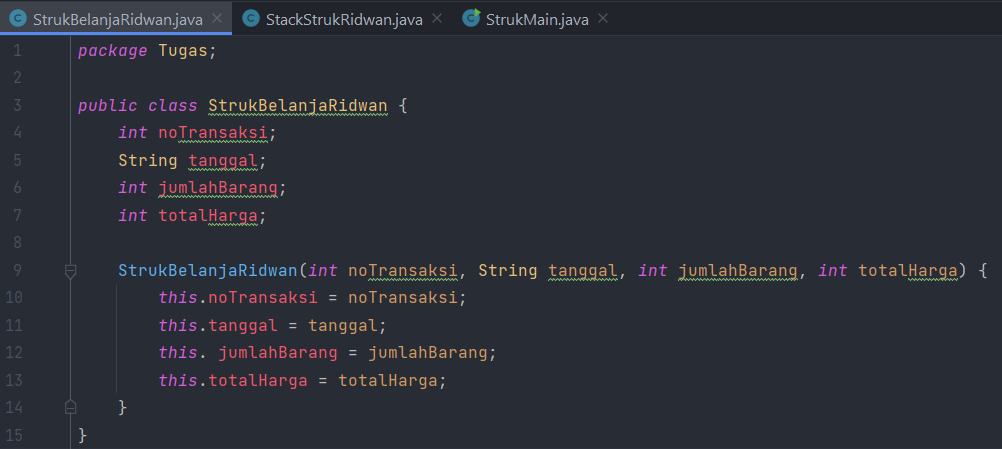
### Output





## Jawaban Nomor 2

### Source Code StrukBelanjaRidwan



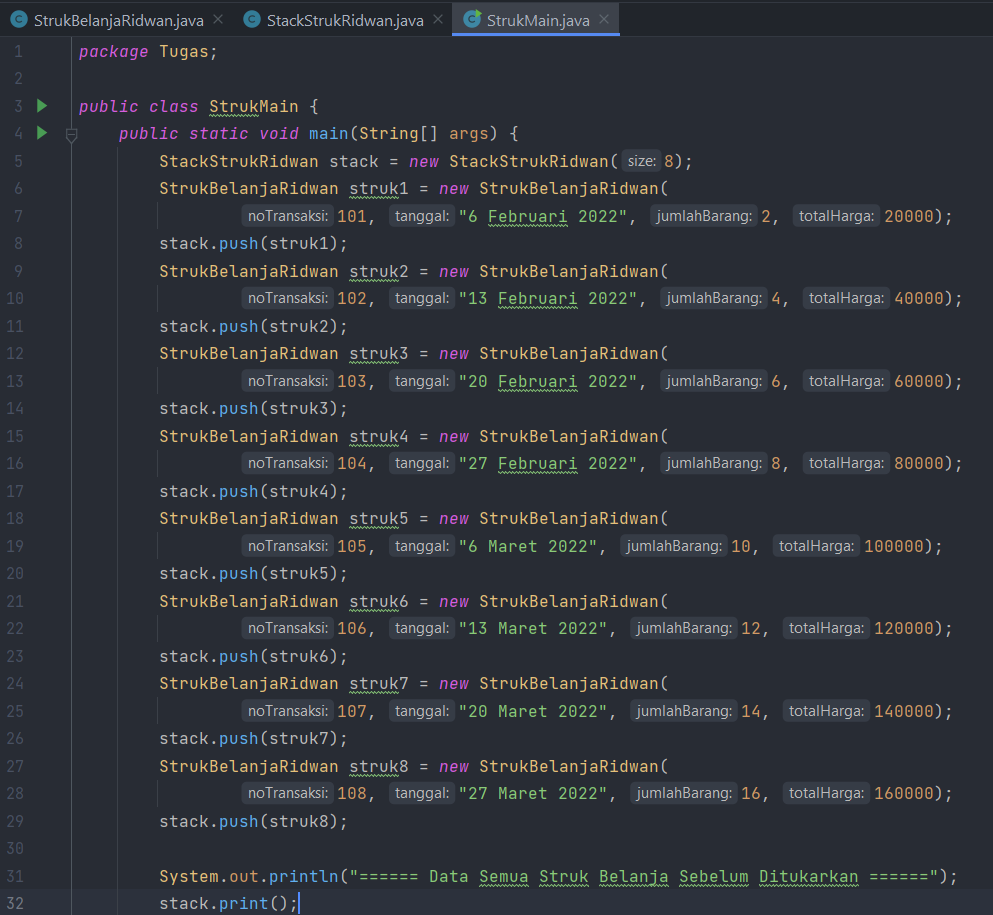
### Source Code StackStrukRidwan

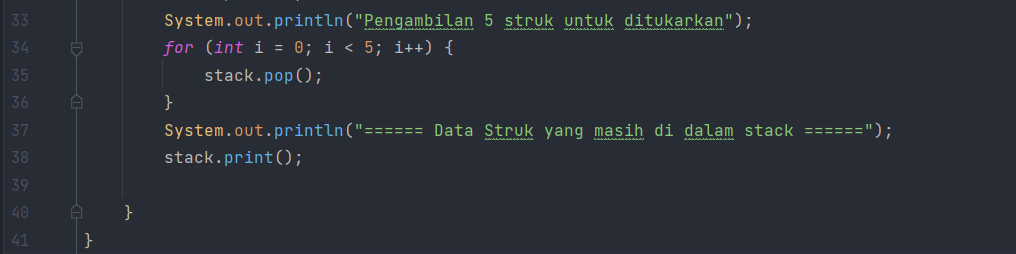






### Source Code StrukMain





### Output

