**Laporan Teori Algoritma dan Struktur Data**

**Jobsheet 9 – Stack**

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



**Nama : Annisa**

**Nim : 2341760032**

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

**Absen : 04**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

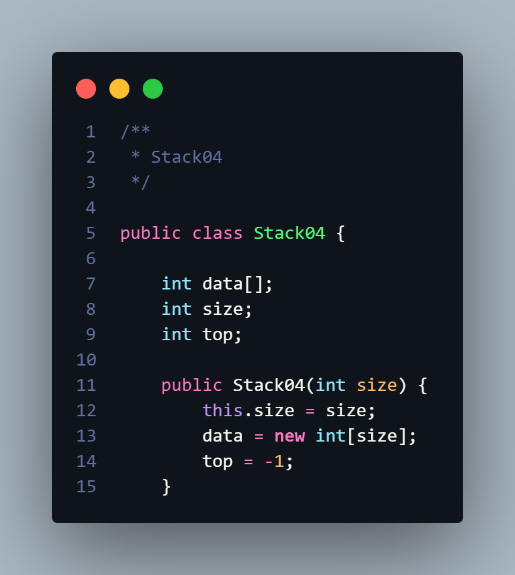
**2023/2024**

**9.2. Praktikum 1**

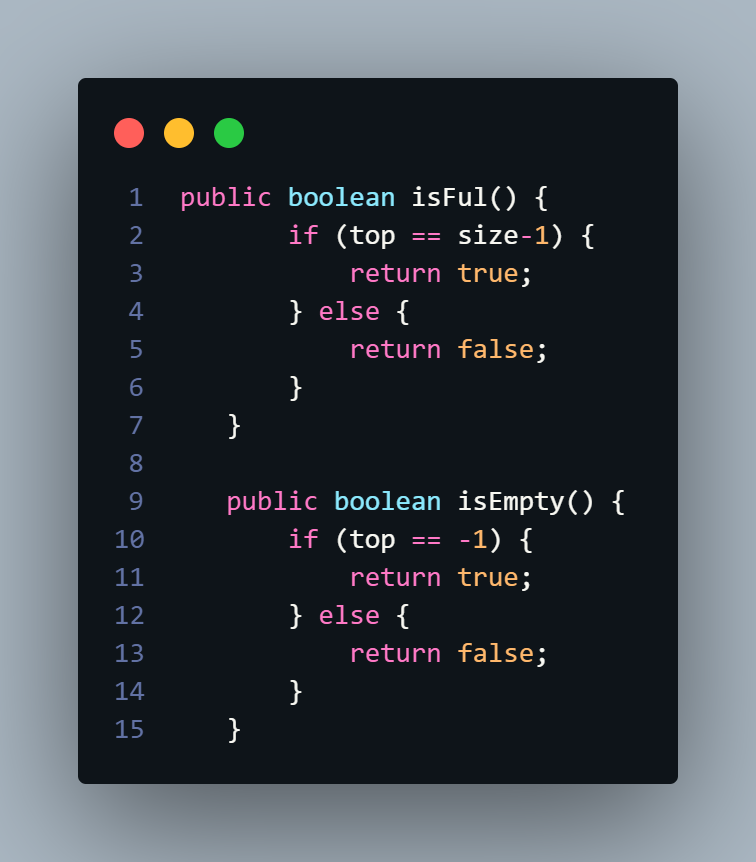
**9.2.1 Langkah-langkah Percobaan**

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

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

****

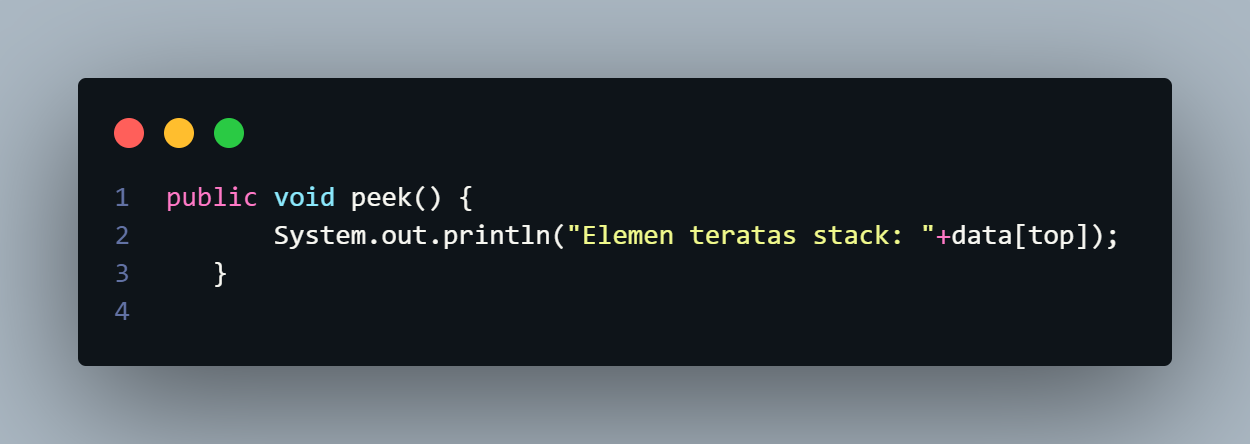
3. Lalu tambahkan method isFull() dan isEmpty() pada class Stack sebagai berikut:

****

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

****

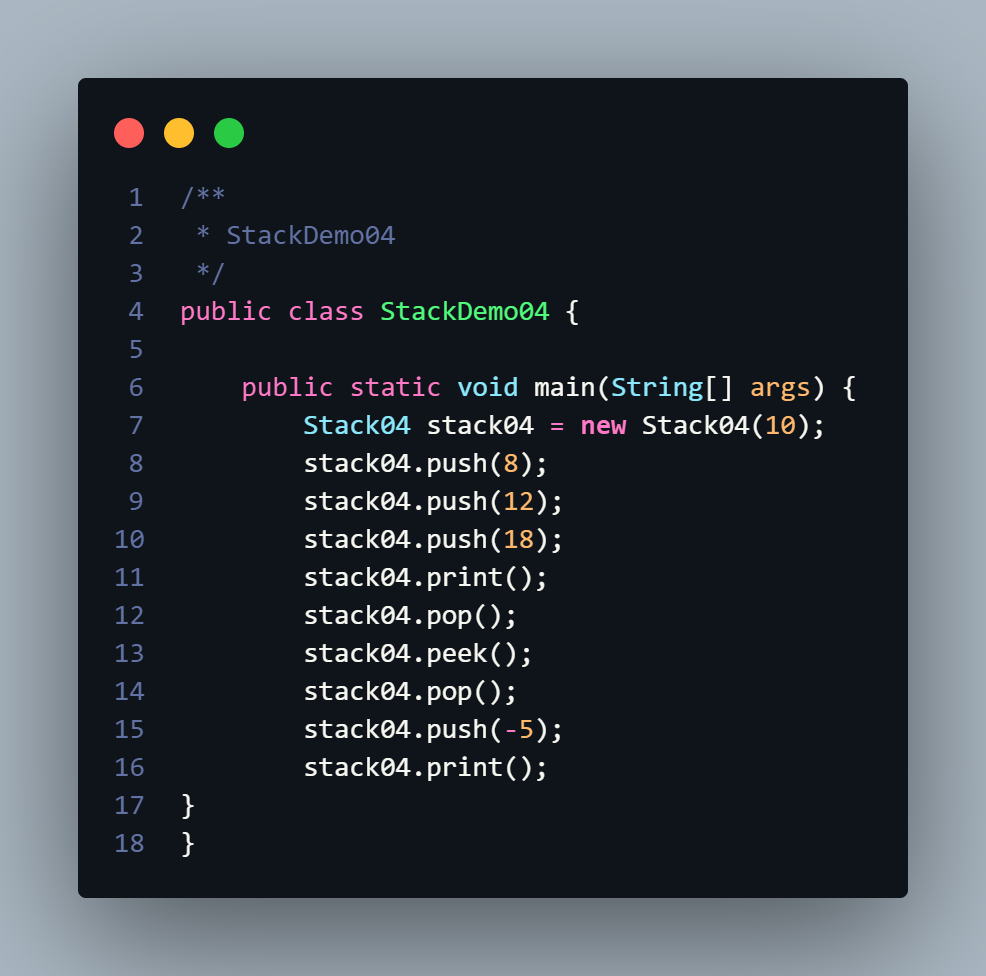
5. Tambahkan method peek() sebagai berikut:

****

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

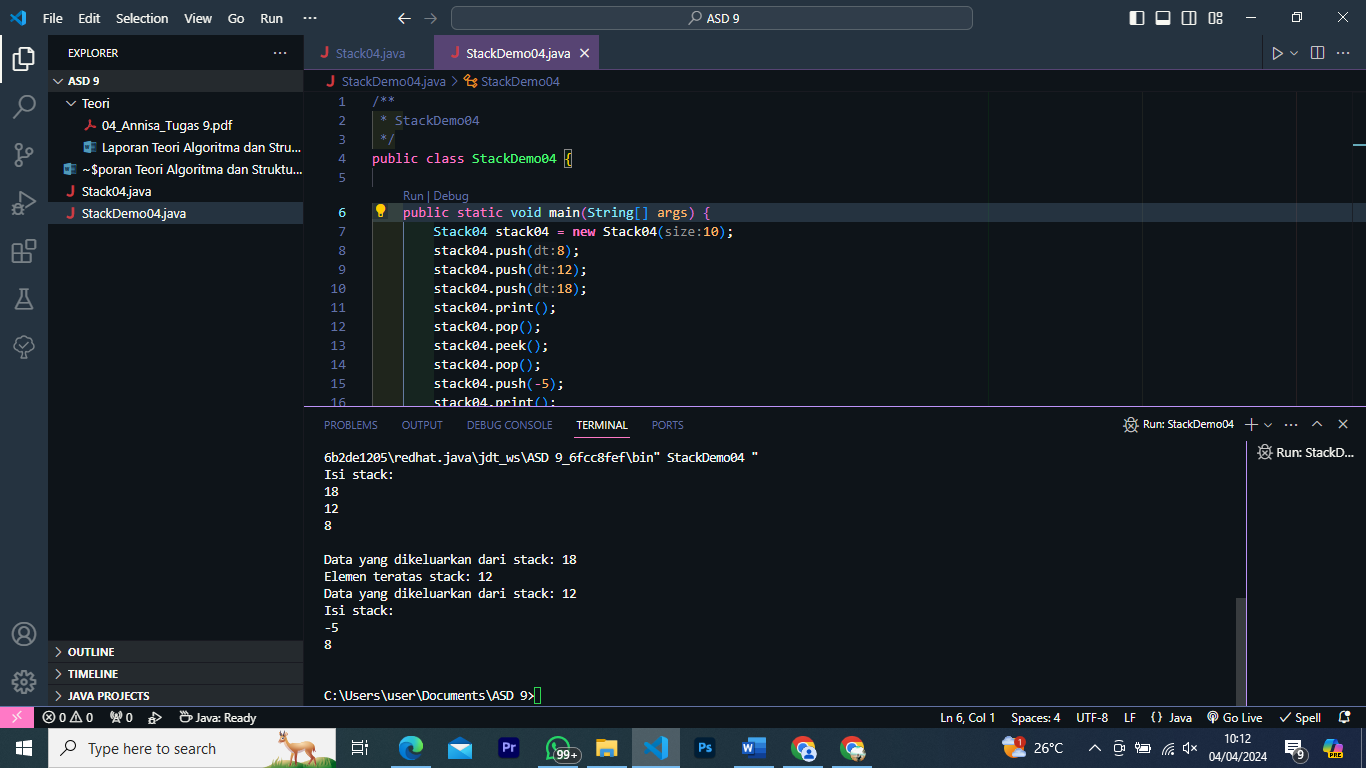


7. Buat file StackDemo.java untuk mengimplementasikan class StackDemo yang berisi fungsi main untuk membuat objek Stack dan mengoperasikan method-method pada class Stack.



8. Compile dan run class StackDemo.

**9.2.2 Verifikasi Hasil Percobaan**



**9.2.3 Pertanyaan**

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

karena untuk menghindari pengecualian. Method isEmpty() digunakan untuk mengecek apakah stack masih kosong atau tidak. Apabila tidak ada pemanggilan method tersebut, maka struktur data mungkin akan mencoba melakukan penghapusan elemen dari struktur data yang kosong. Ini bisa menyebabkan kesalahan atau pengecualian, seperti "IndexError" atau "NoSuchElementException".

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

metode peek() digunakan untuk melihat elemen teratas dari stack tanpa menghapusnya. Saat memanggil peek(), elemen teratas dari stack hanya dilihat dan tidak dihapus dari stack. Metode pop() digunakan untuk menghapus dan mengembalikan elemen teratas dari stack. Saat Anda memanggil pop(), elemen teratas stack dihapus dari stack, dan nilai tersebut dikembalikan atau digunakan dalam operasi selanjutnya.

**9.3. Praktikum 2**

**9.3.1. Langkah-langkah Percobaan**

1. Perhatikan Diagram Class Pakaian berikut ini:

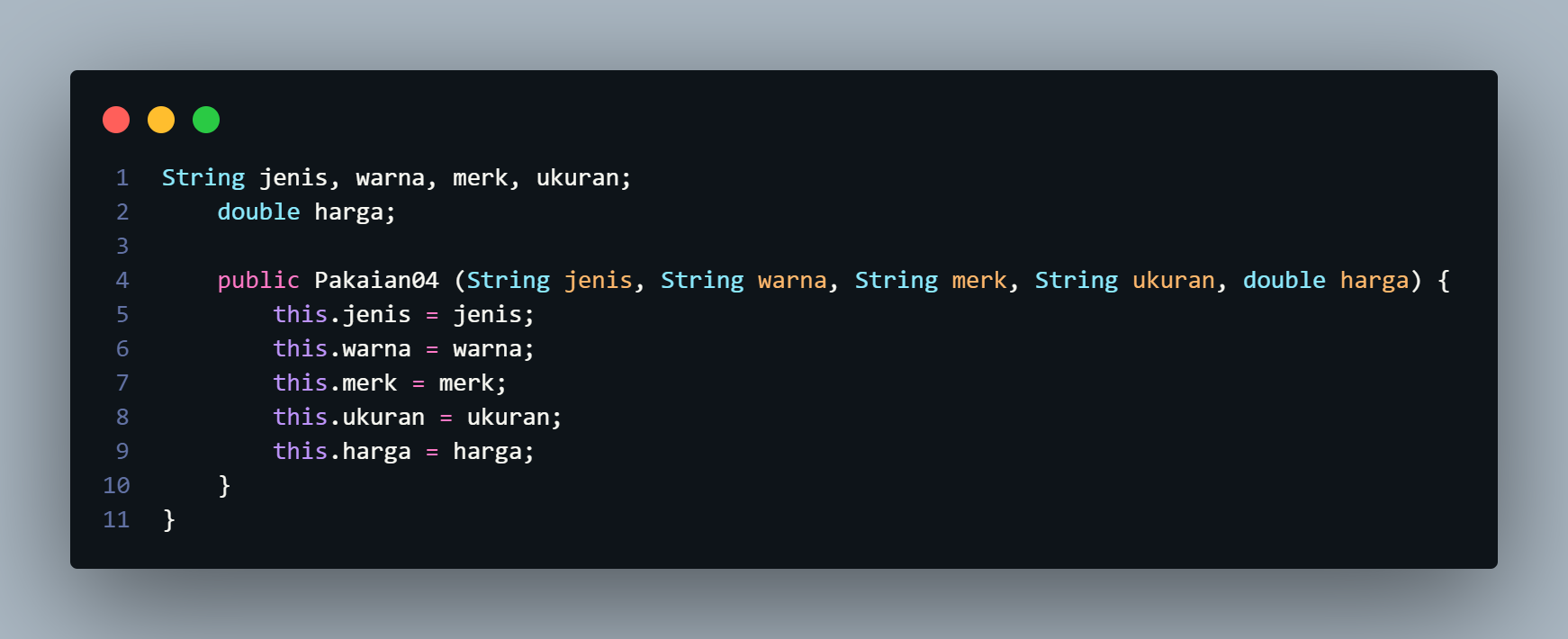
****

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

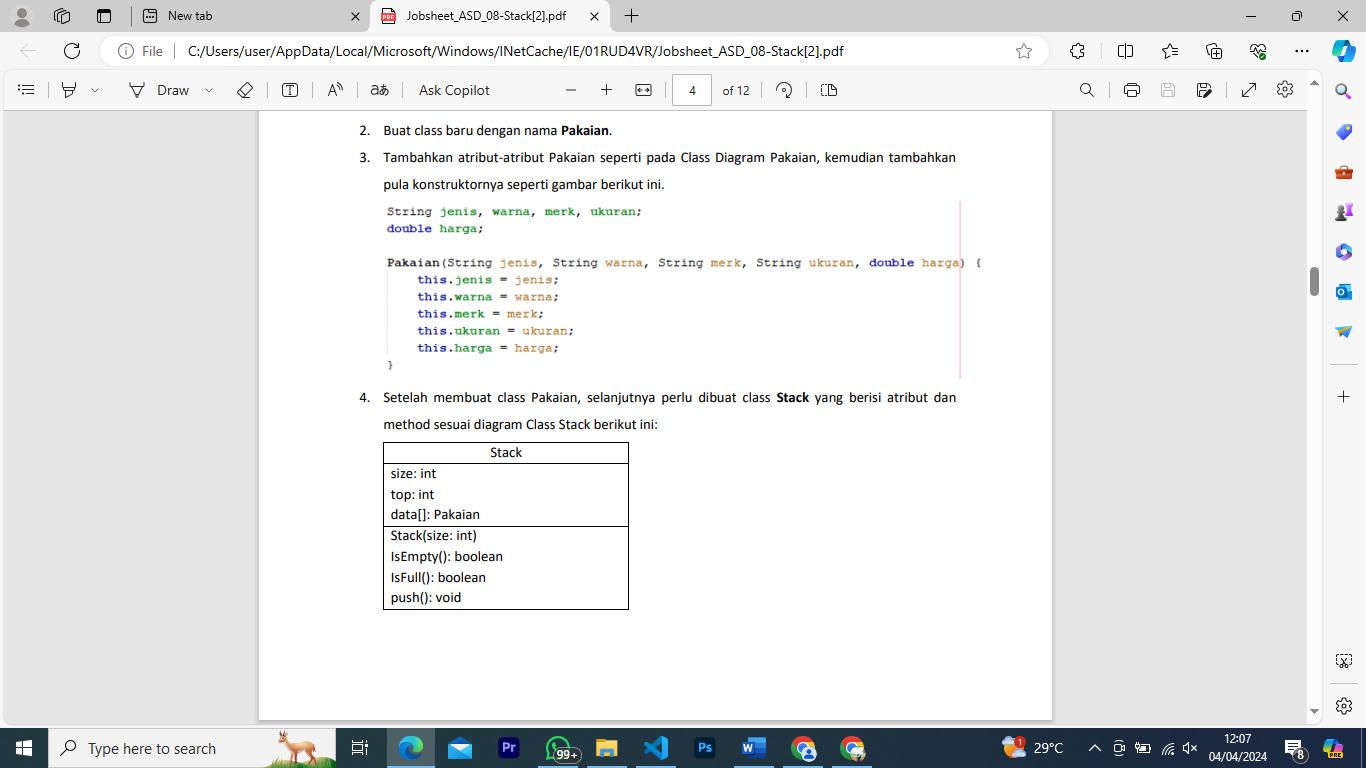
2. Buat class baru dengan nama Pakaian.

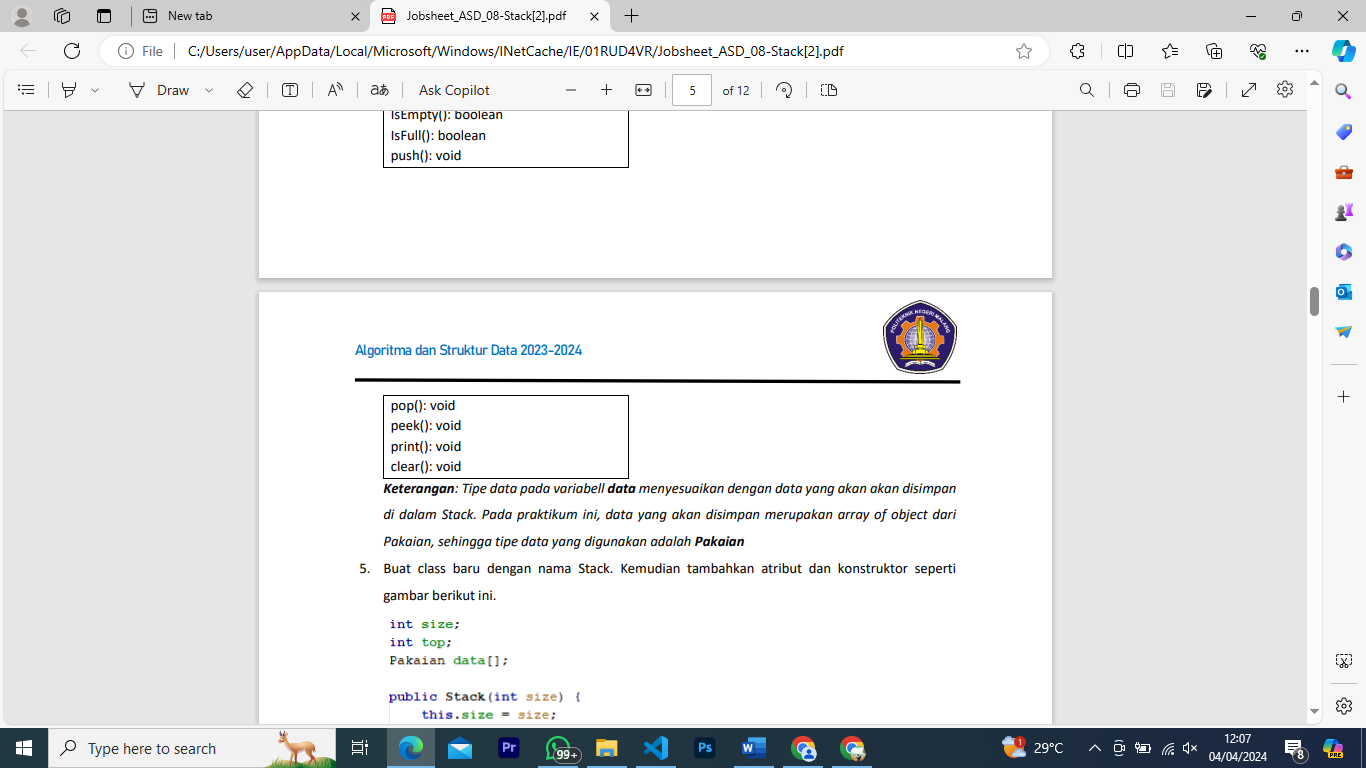
****

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



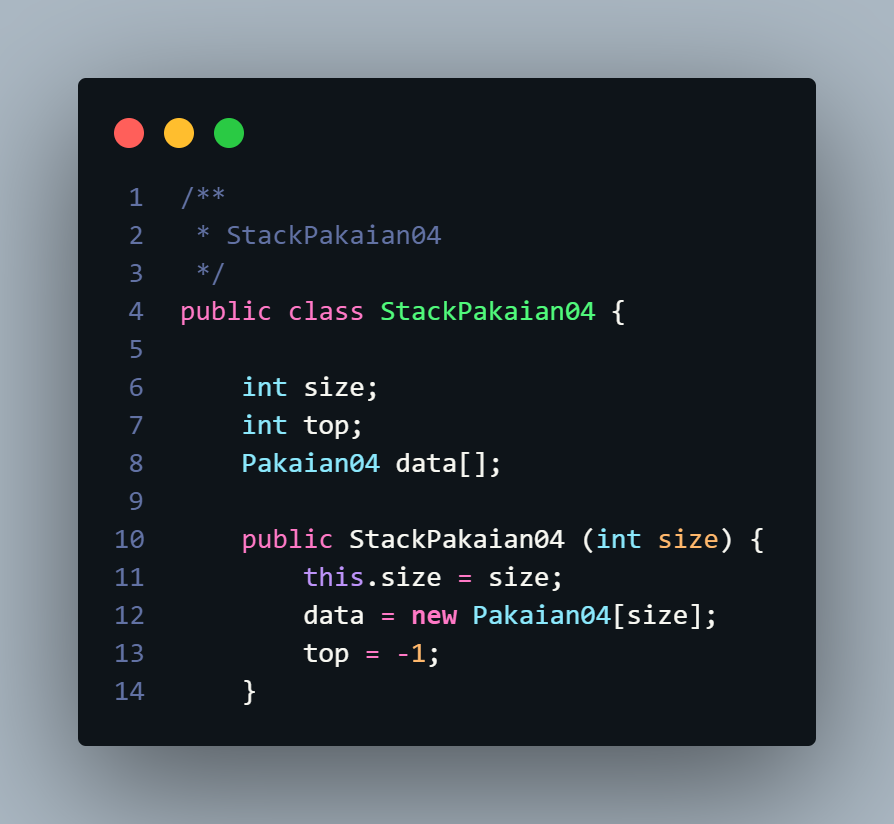
4. Setelah membuat class Pakaian, selanjutnya perlu dibuat class Stack yang berisi atribut dan method sesuai diagram Class Stack berikut ini:



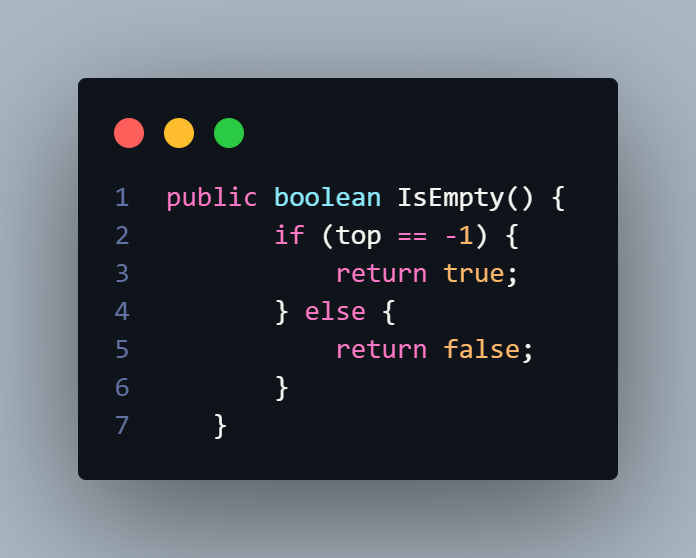


Keterangan: Tipe data pada variabell data menyesuaikan dengan data yang akan akan disimpan di dalam Stack. Pada praktikum ini, data yang akan disimpan merupakan array of object dari Pakaian, sehingga tipe data yang digunakan adalah Pakaian

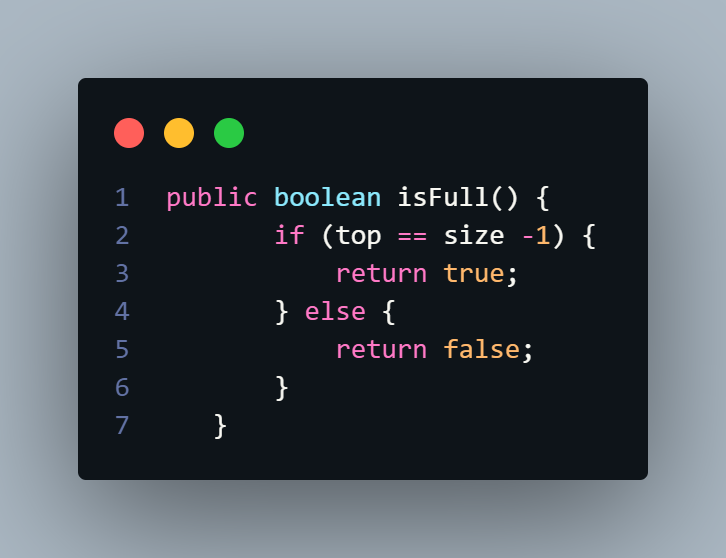
5. Buat class baru dengan nama Stack. Kemudian tambahkan atribut dan konstruktor seperti gambar berikut ini.

****

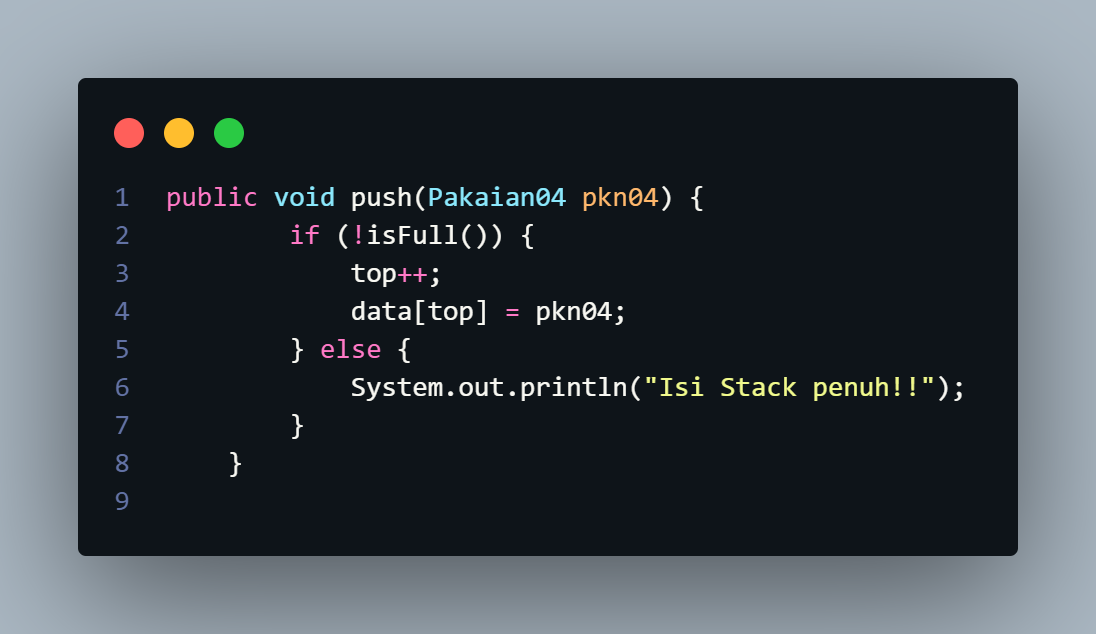
6. Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah stack kosong.

****

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

****

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



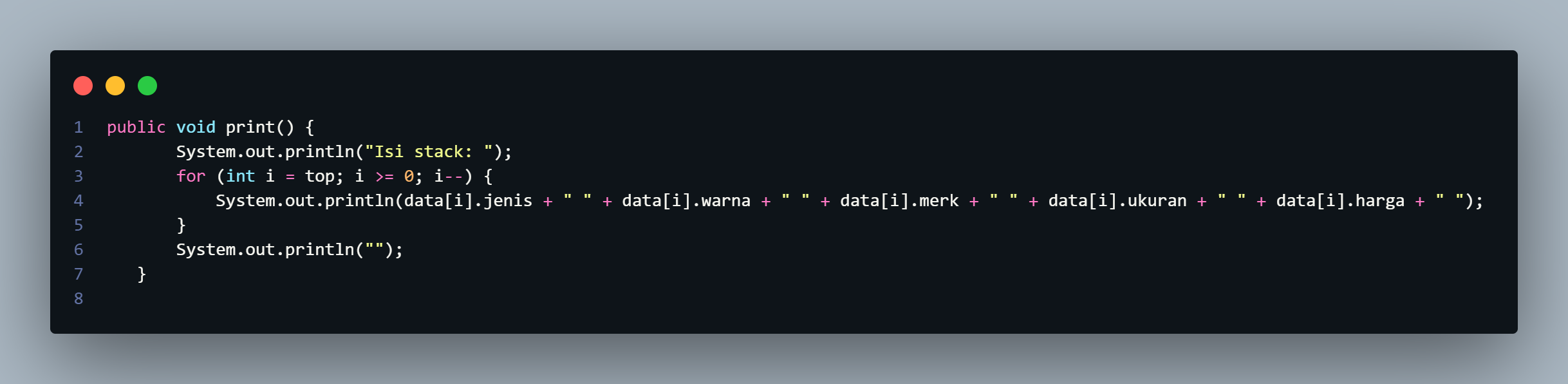
9. Buat method Pop bertipe void untuk mengeluarkan isi elemen stack. Karena satu elemen stack terdiri dari beberapa informasi (jenis, warna, merk, ukuran, dan harga), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut



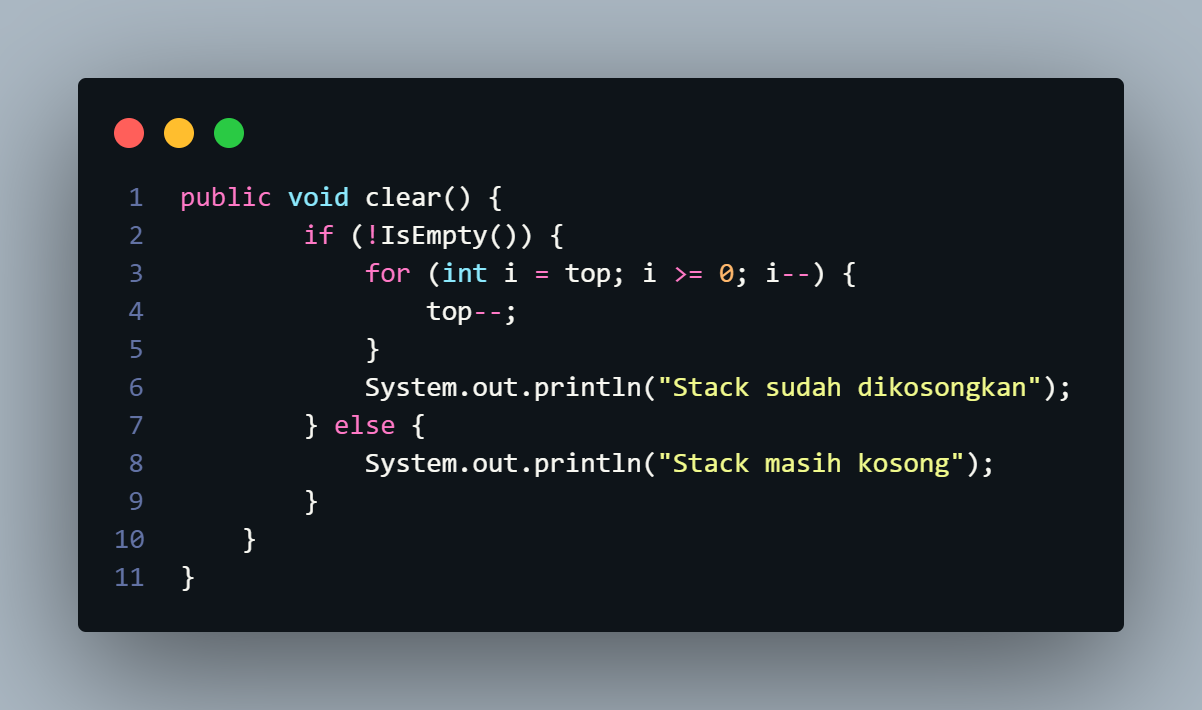
10. Buat method peek bertipe void untuk memeriksa elemen stack pada posisi paling atas.



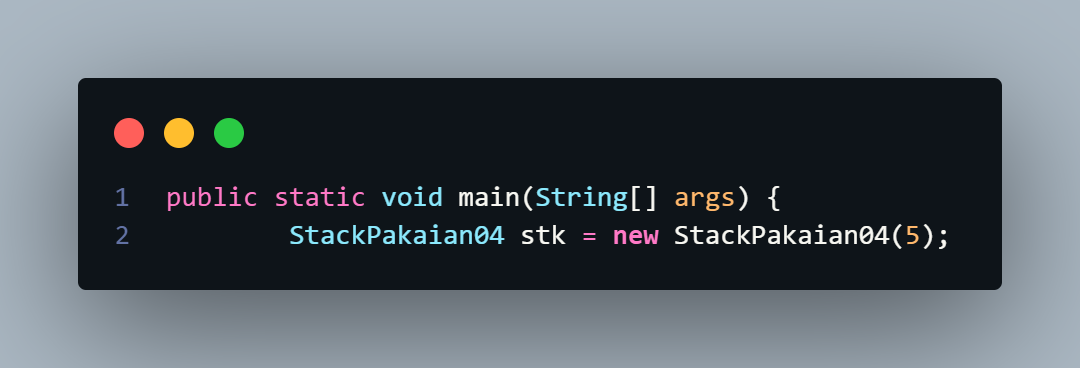
11. Buat method print bertipe void untuk menampilkan seluruh elemen pada stack.

****

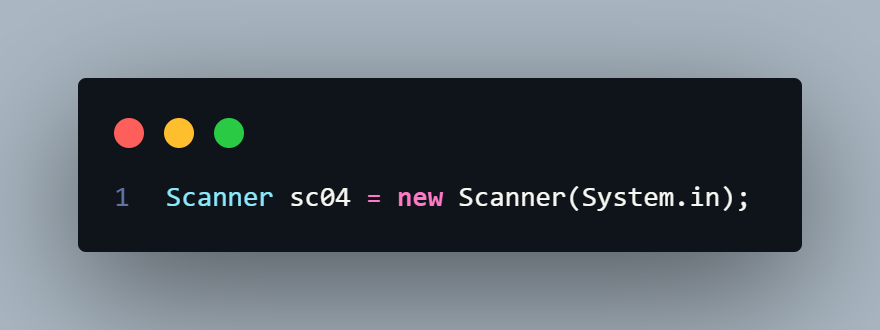
12. Buat method clear bertipe void untuk menghapus seluruh isi stack.

****

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.

****

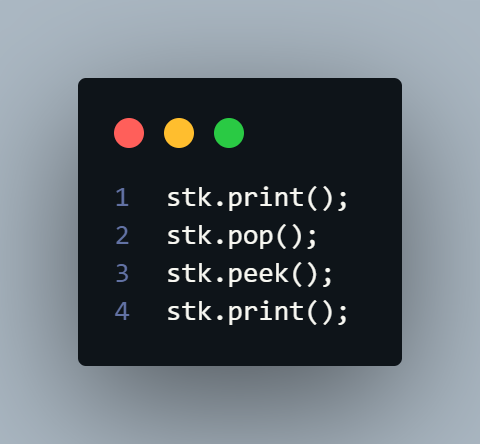
14. Deklarasikan Scanner dengan nama sc

****

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

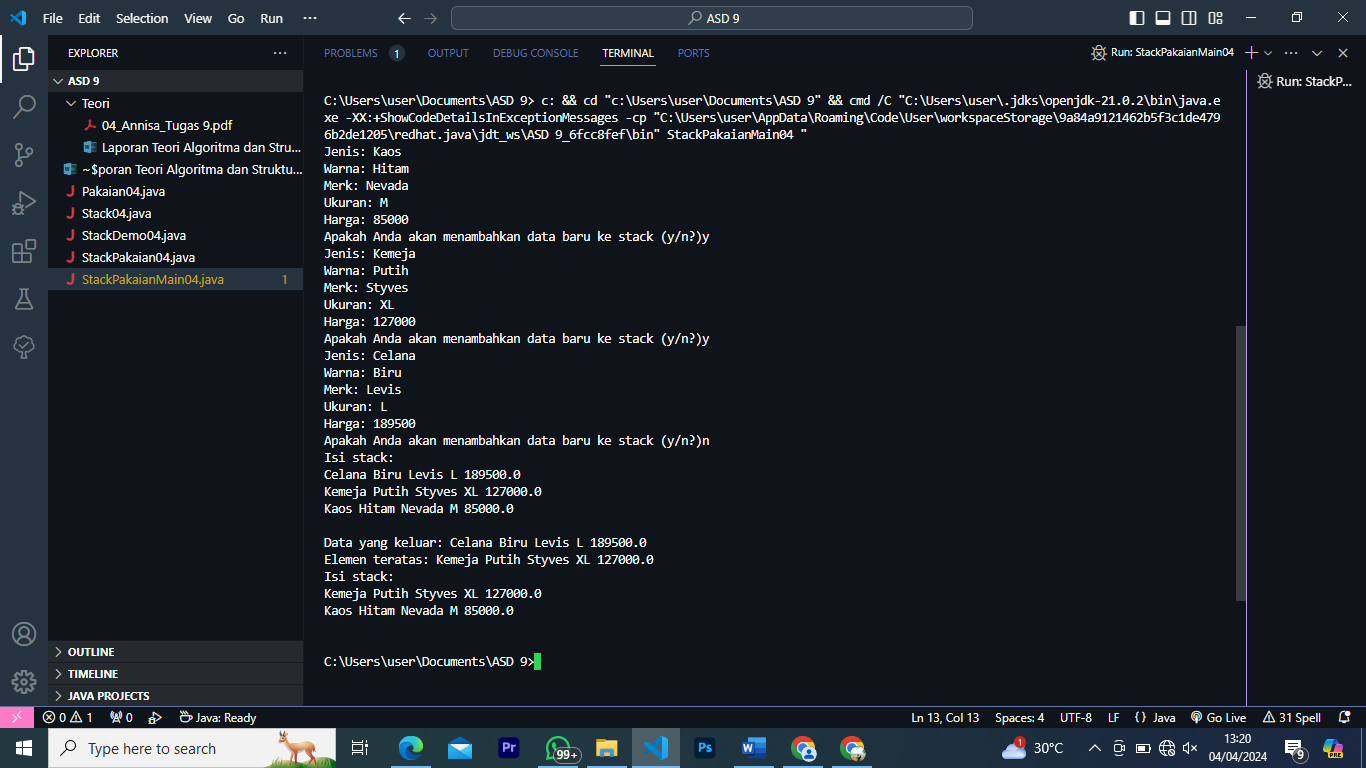
****

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

****

17. Compile dan jalankan class StackMain, kemudian amati hasilnya.

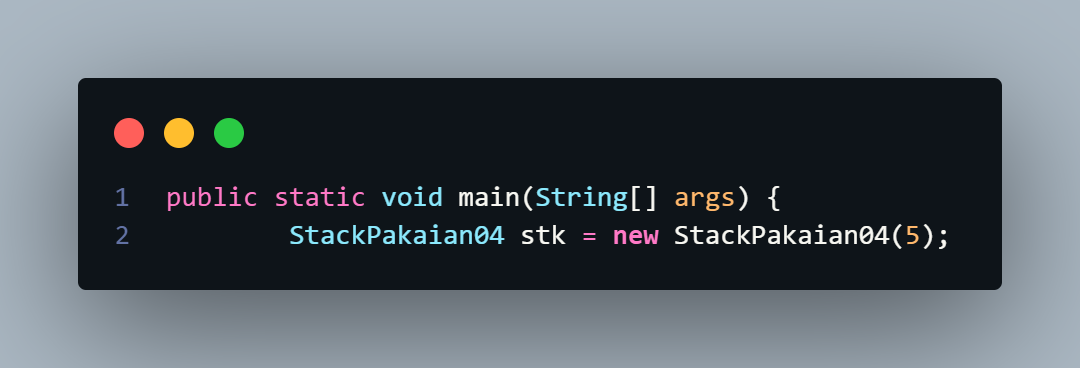
**9.3.2. Verifikasi Hasil Percobaan**



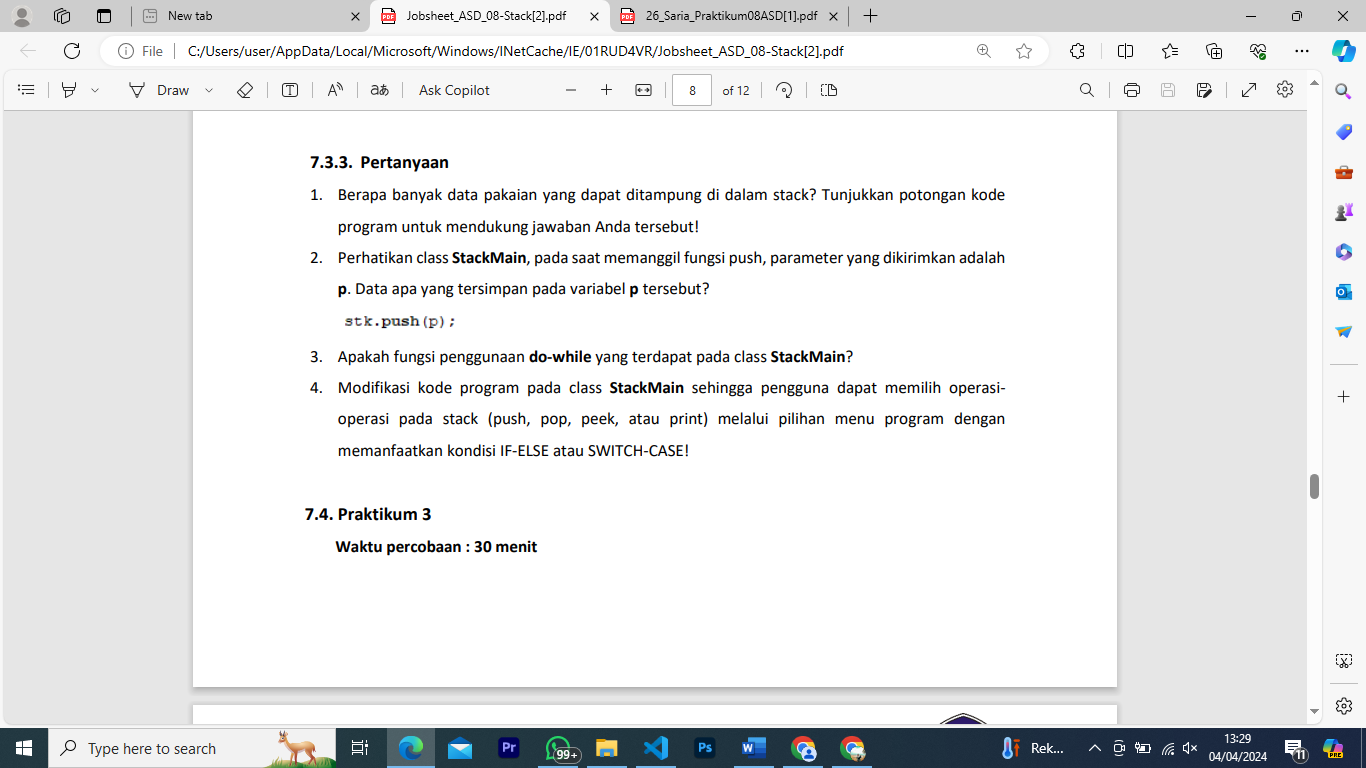
**9.3.3. Pertanyaan**

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

Banyak data pakaian yang dapat ditampung di dalam stack berjumlah 5



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

****

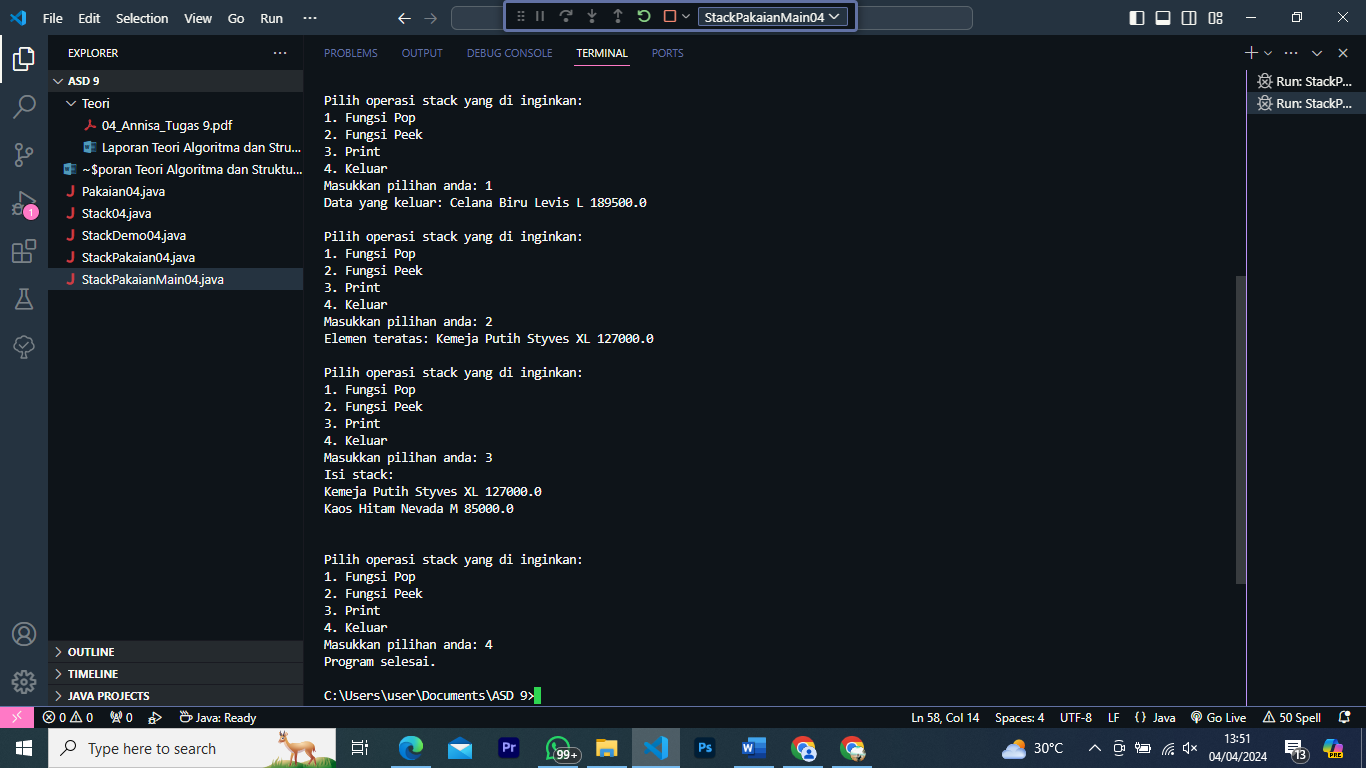
Data yang tersimpan pada variable p adalah data-data yang dimasukkan melalui Scanner

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

Berfungsi untuk melakukan perulangan agar nantinya user dapat menginputkan data yang meliputi judul, nama pengarang, tahun terbit, jumlah halaman, dan harga. Maka setelah melakukan penginputan, user akan diberi pilihan untuk menambahkan data baru atau tidak, jika "y" maka user akan diminta untuk menginputkan data data lagi, dan jika user memilih "n" maka program akan berhenti.

**4. 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!**

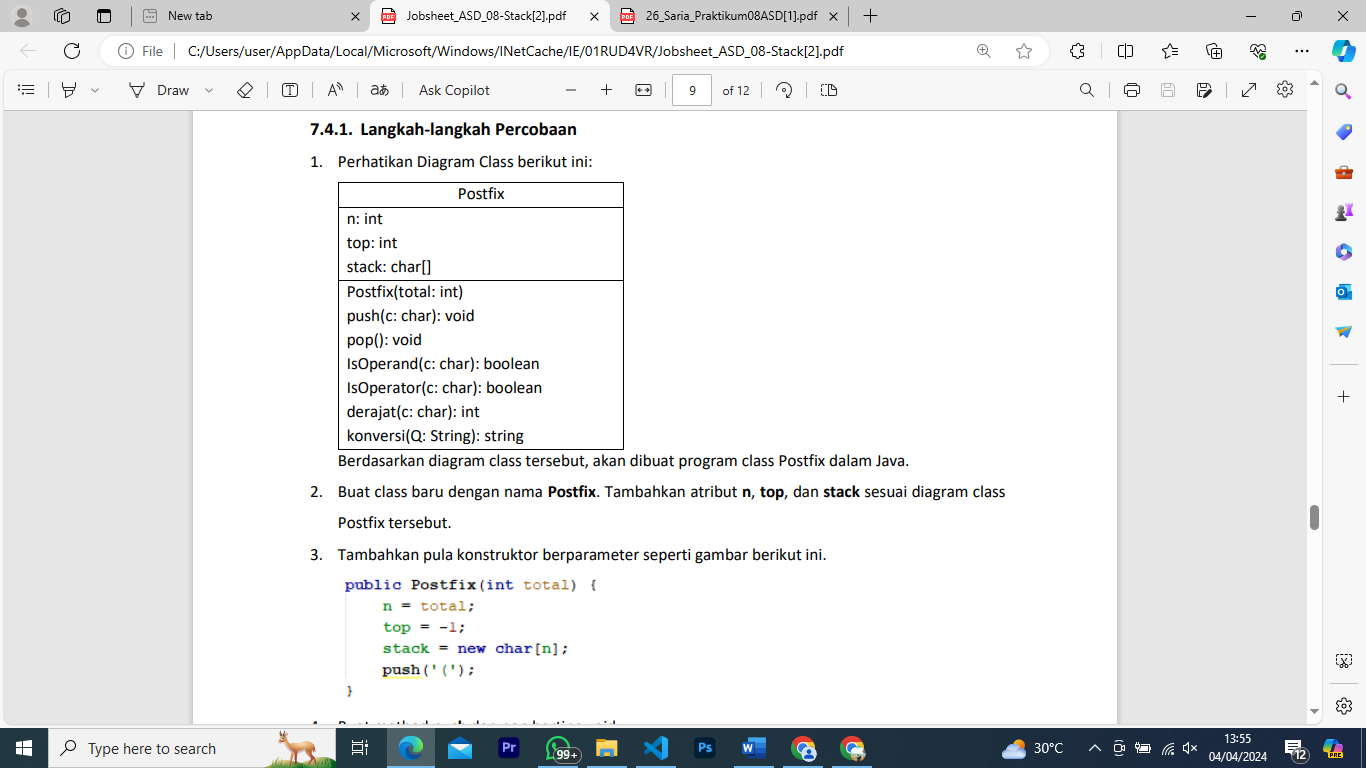
Berikut merupakan hasil modifikasi kode program

**9.4. Praktikum 3**

**9.4.1. Percobaan**

1. Perhatikan Diagram Class berikut ini:



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

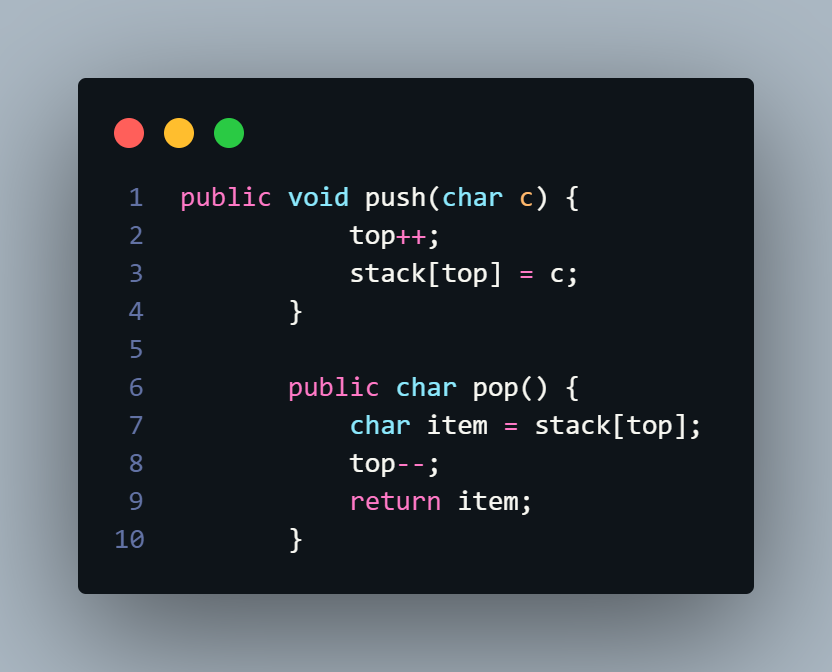
2. Buat class baru dengan nama Postfix. Tambahkan atribut n, top, dan stack sesuai diagram class Postfix tersebut.

****

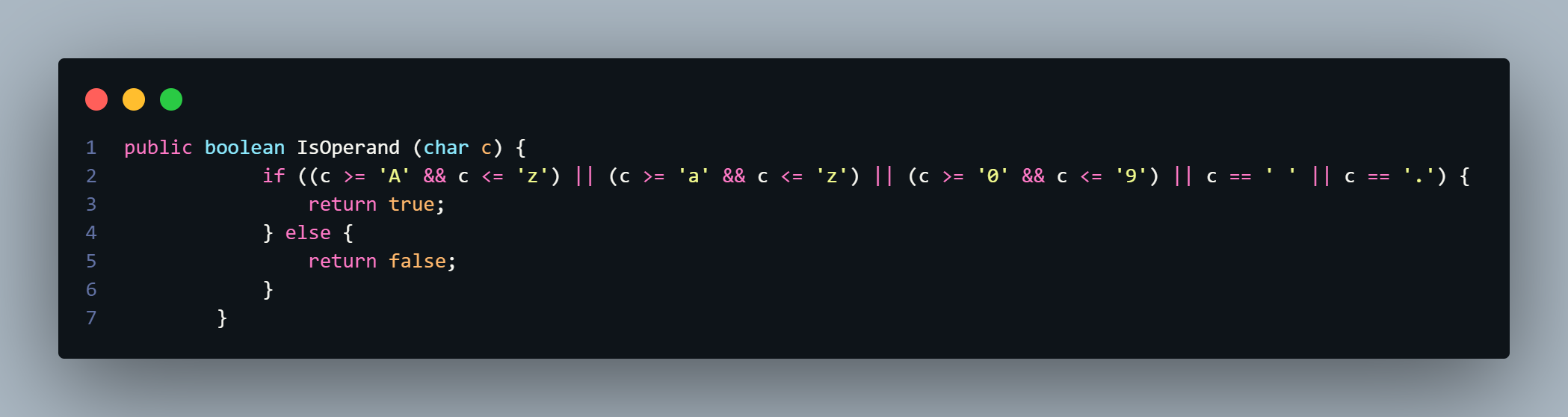
3. Tambahkan pula konstruktor berparameter seperti gambar berikut ini.

****

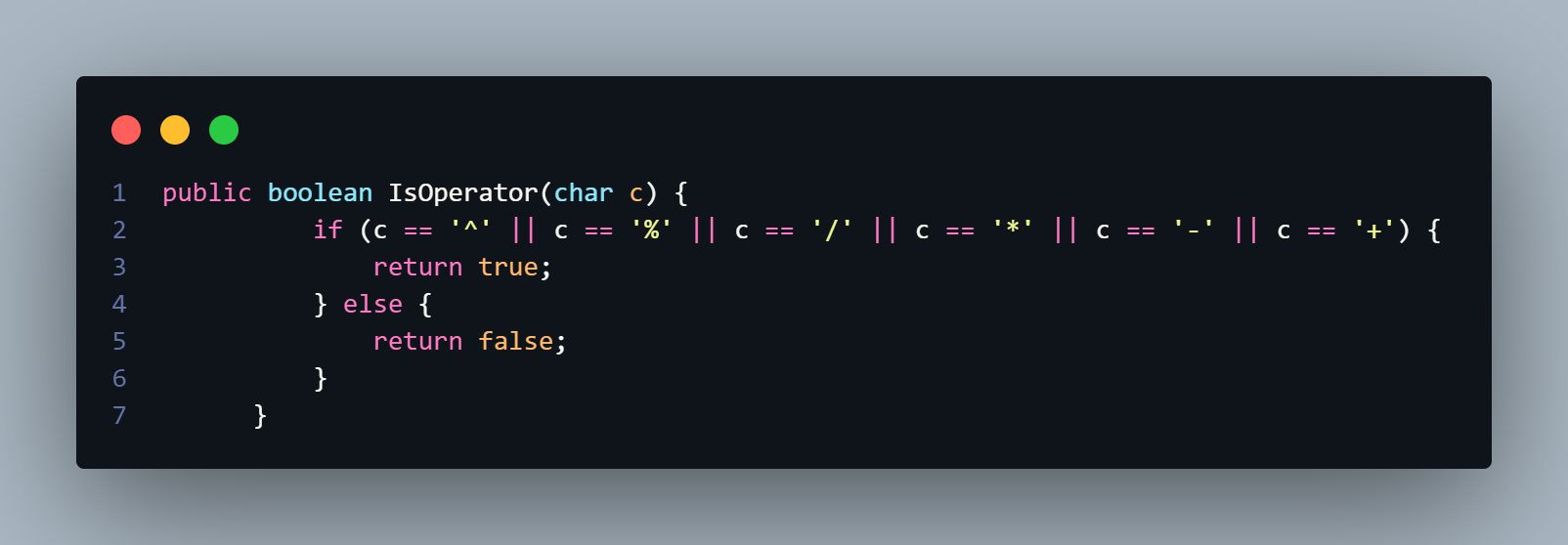
4. Buat method push dan pop bertipe void.



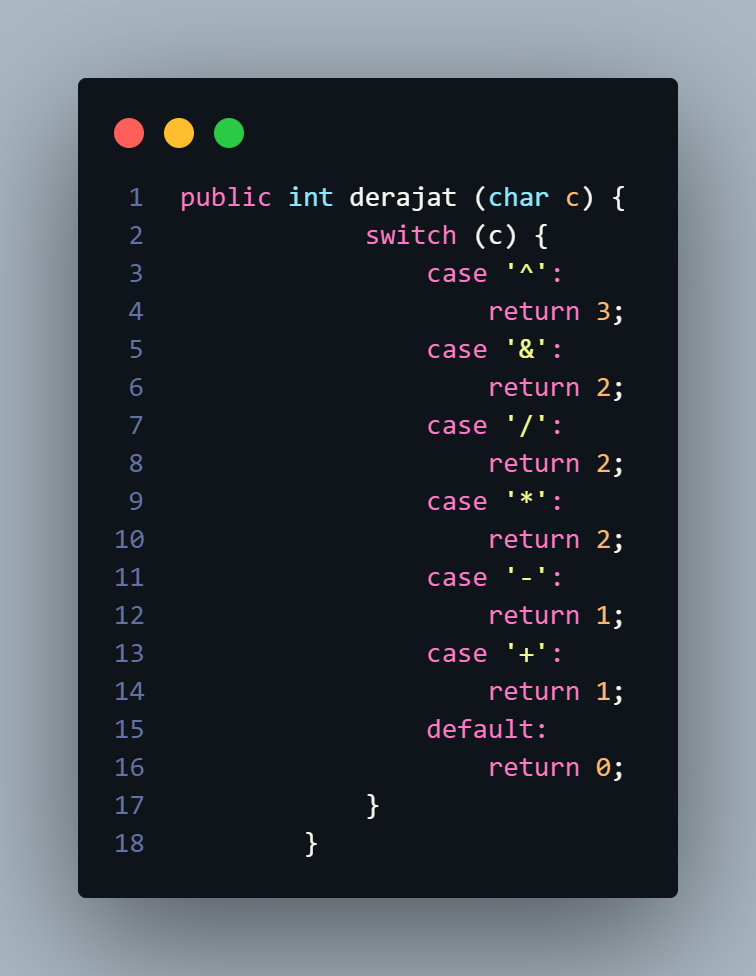
5. Buat method IsOperand dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operand



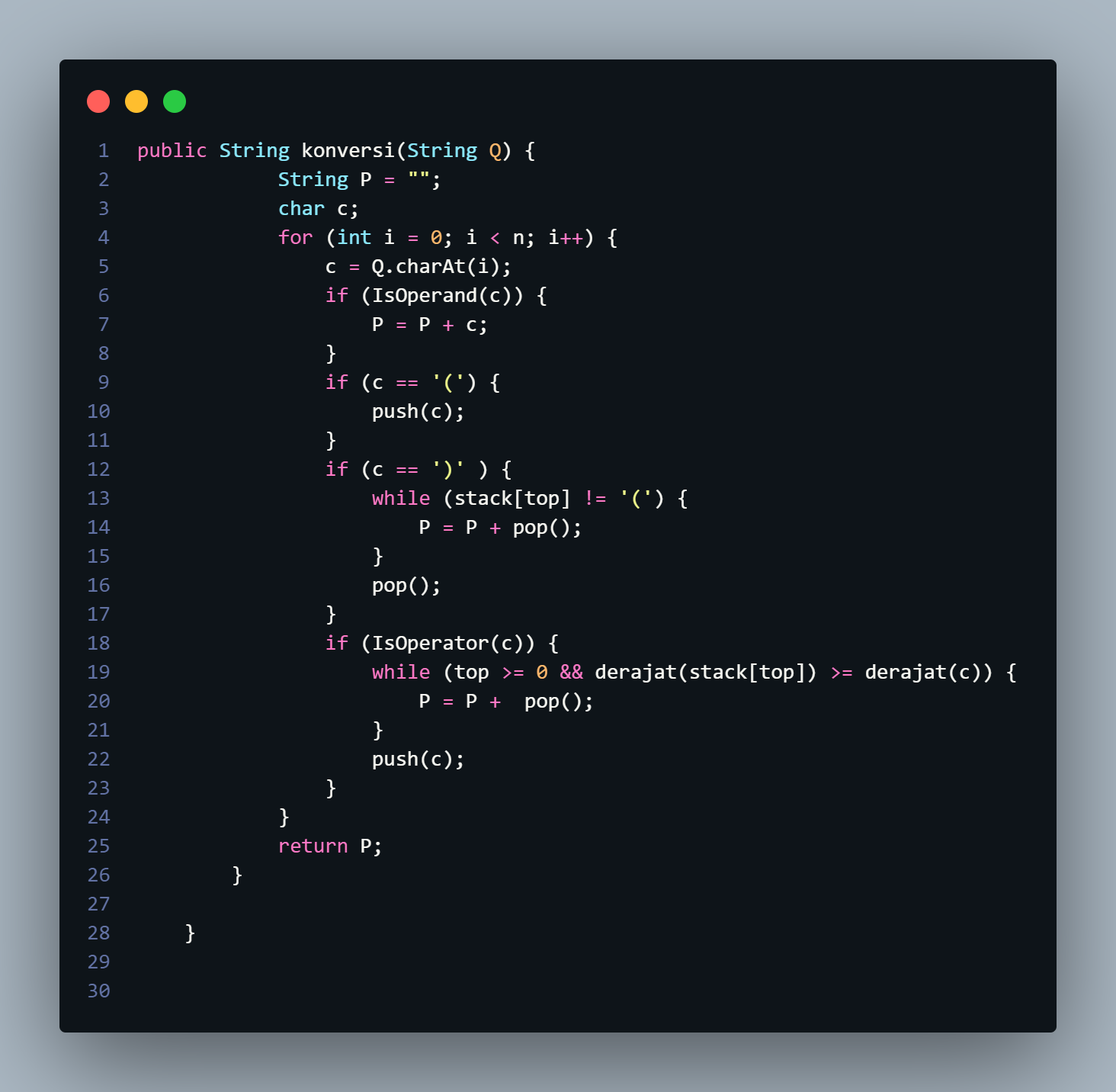
6. Buat method IsOperator dengan tipe boolean yang digunakan untuk mengecek apakah elemen data berupa operator.



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



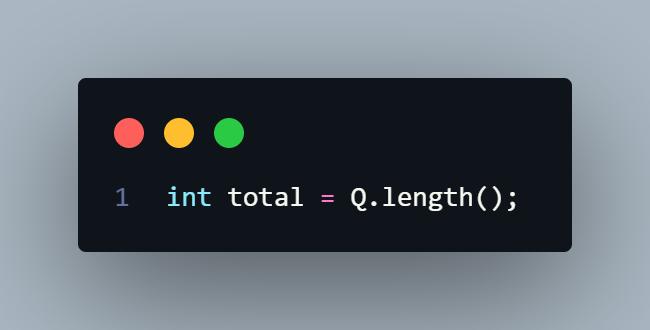
8. Buat method konversi untuk melakukan konversi notasi infix menjadi notasi postfix dengan cara mengecek satu persatu elemen data pada String Q sebagai parameter masukan.



9. 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.



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

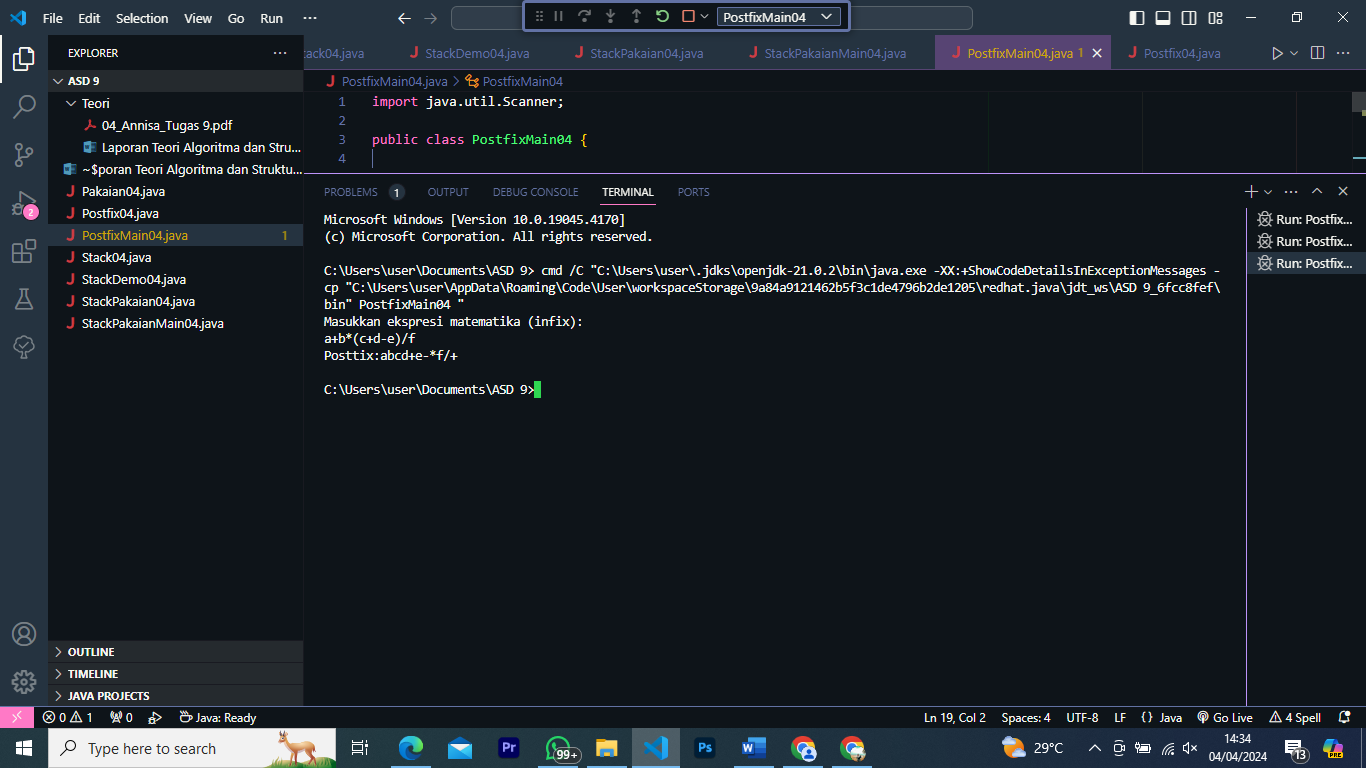
****

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.

****

12. Compile dan jalankan class PostfixMain dan amati hasilnya.

**9.4.2. Verifikasi Hasil Percobaan**

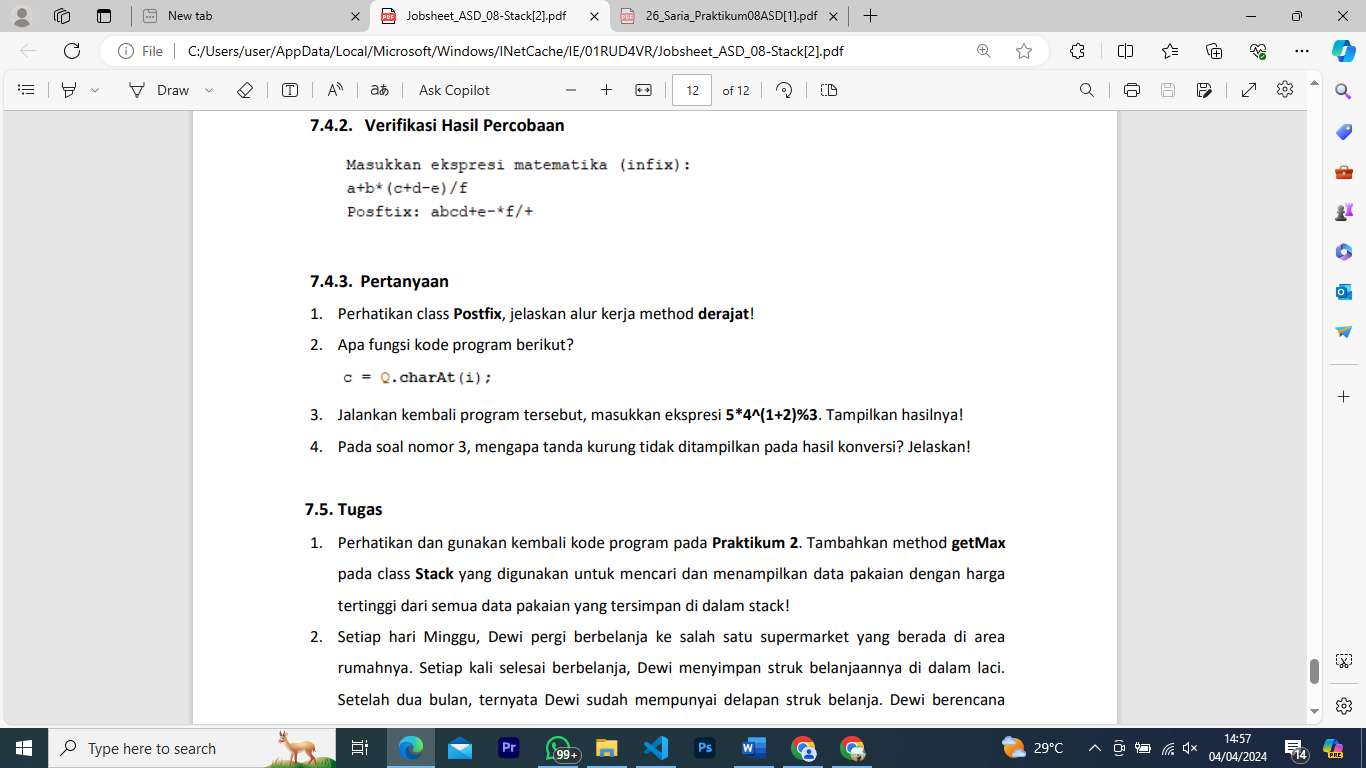
****

**9.4.3. Pertanyaan**

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

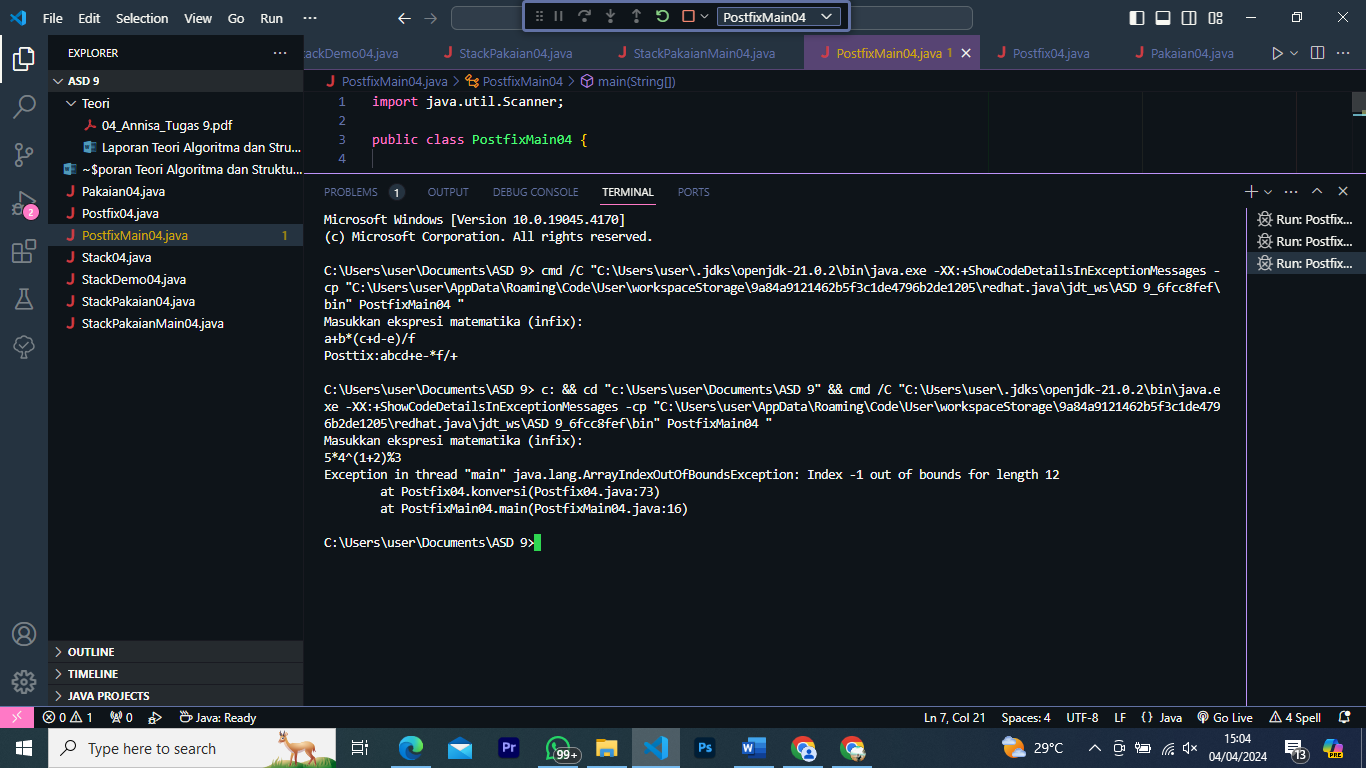
Pada class "postix", method derajat memiliki alur seperti berikut ini ketika operator tersebut adalah '^' maka akan mereturn nilai 3, jika operator '%','/','\*' maka akan mereturn nilai 2. dan sedangkan operator '-', '+' maka akan mereturn nilai 1 dengan menggunakan perintah switch case yang telah tersedia.

**2. Apa fungsi kode program berikut?**

****

Fungsi kode program tersebut yaitu Variabel c digunakan untuk menyimpan data char i ke dalam variabel Q dengan menggunakan perintah charAt(i).

**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!**

****

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

Karena ketika terdapat tanda kurung pada ekspresi matematika, maka tanda kurung tersebut akan langsung di pop tanpa dimasukkan kedalam postfix.

**9.5. Tugas**

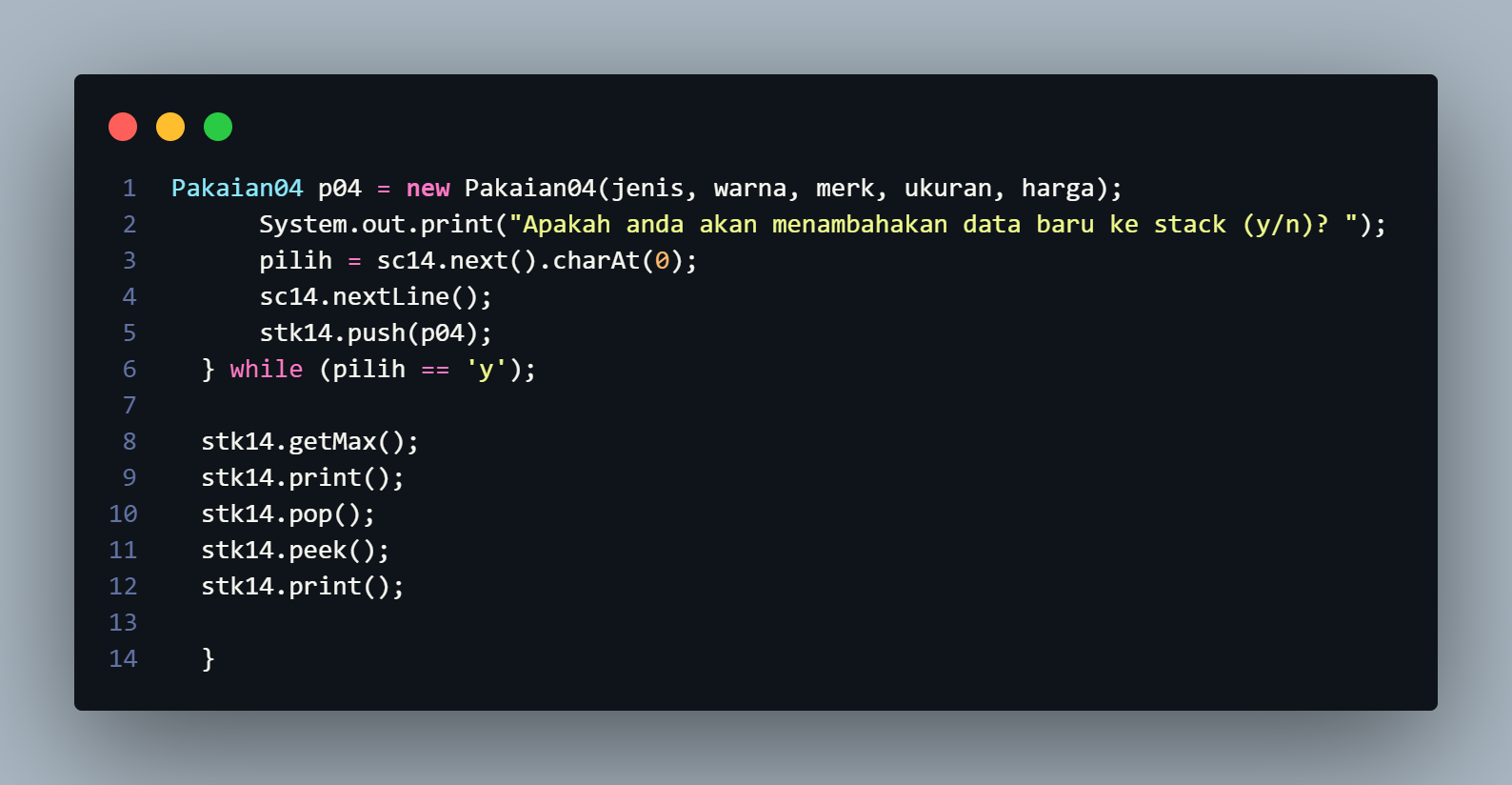
**1. Perhatikan dan gunakan kembali kode program pada Praktikum 2. Tambahkan method getMax pada class Stack yang digunakan untuk mencari dan menampilkan data pakaian dengan harga tertinggi dari semua data pakaian yang tersimpan di dalam stack!**

Berikut merupakan kode program yang telah dimodifikasi dari praktikum 2

Gambar di bawah merupakan getMax yang terdapat pada class StackPakaian04



Gambar di bawah merupakan pemanggilan method getMax pada class StackPakaianMain04



Gambar di bawah ini merupakan hasil running dari praktikum 2 yang telah dimodifikasi



**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 Tampilkan informasi struk belanja yang masih tersimpan di dalam stack**

Berikut merupakan kode program dari tugas 2

****

****

Berikut merupakan hasil running dari kode program di atas





