**LAPORAN PRAKTIKUM JOBSHEET 2**

**KONSEP DASAR PEMROGRAMAN**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**



**Disusun Oleh :**

**Jami’atul Afifah (2341760102)**

**SIB-1F**

**PROGRAM STUDI D4 SISTEM INFOEMASI BISNIS**

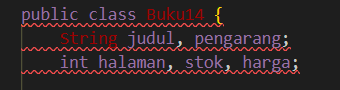
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

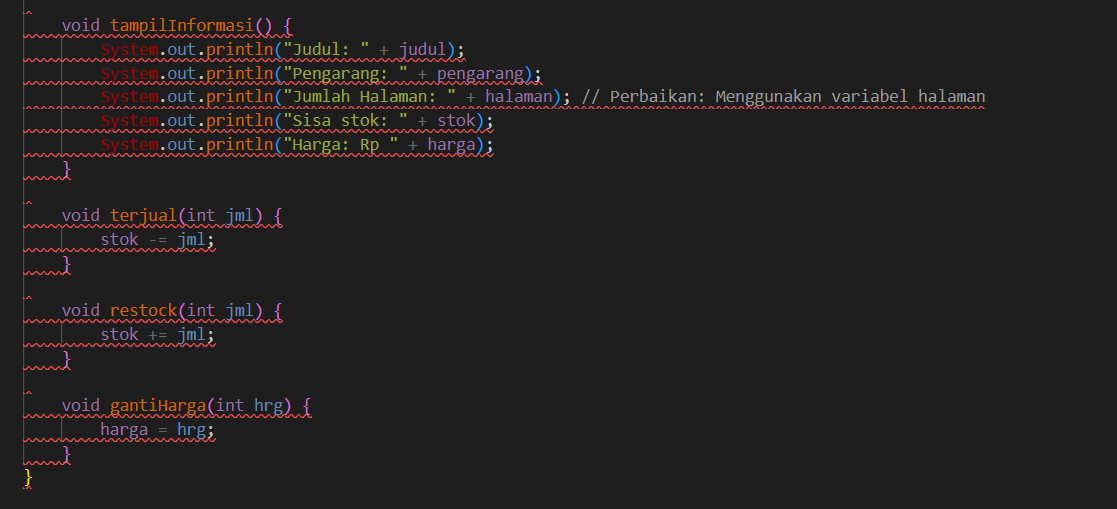
**2024**

**Percobaan 1: Deklarasi Class, Atribut dan Method**

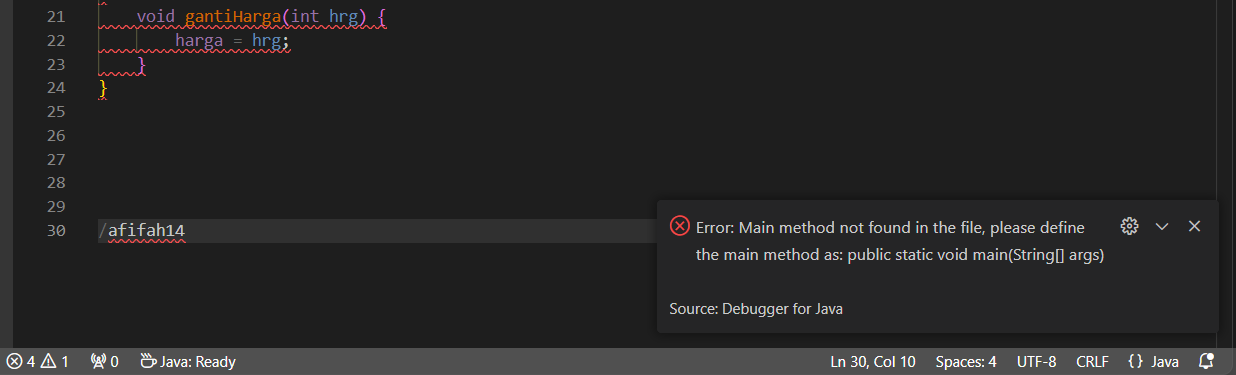
Tambahkan variable A group of black and blue letters  Description automatically generated



Buat class buku dan tambahkan method



Sehingga Ketika di run maka hasilnya seperti ini



**Pertanyaan**

1. Sebutkan dua karakteristik class atau object!

* Encapsulation: Kemampuan untuk menyembunyikan rincian implementasi internal dari objek dan hanya mengekspos fungsionalitas publiknya.
* Inheritance: Kemampuan sebuah kelas untuk mewarisi sifat dan perilaku dari kelas lainnya, yang memungkinkan penggunaan kembali kode dan membuat hubungan hierarki antara kelas

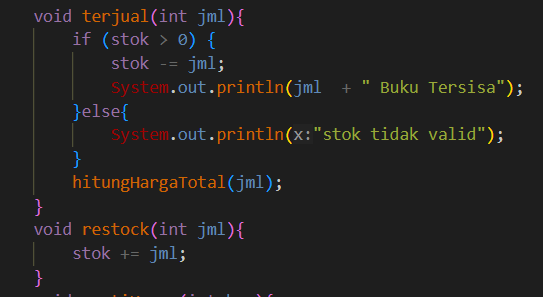
1. Perhatikan class **Buku** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!

Ada 5 yaitu judul, pengarang, halaman, stok, dan harga

1. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya

Ada 4 method yaitu tampilInformasi(), terjual(int jml), restock(int jml), gantiHarga(int hrg)

1. Perhatikan method **terjual()** yang terdapat di dalam class **Buku**. Modifikasi isi method tersebut sehingga proses pengurangan hanya dapat dilakukan jika stok masih ada (lebih besar dari 0)!

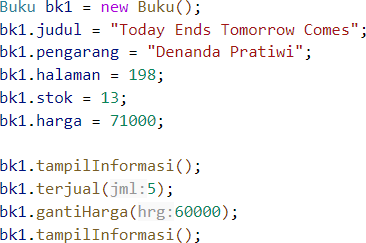


1. Menurut Anda, mengapa method **restock()** mempunyai satu parameter berupa bilangan int?

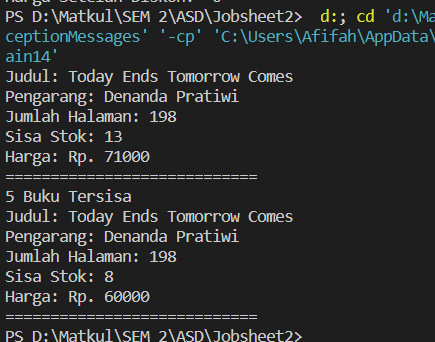
untuk menentukan jumlah tambahan stok yang akan ditambahkan

**Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method**

Buat file baru BukuMain14.java dan didalam fungsi main() lakukan instansi



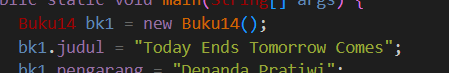
Sehingga apabila dijalankan hasilnya

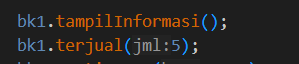


**Pertayaan**

1. Pada class BukuMain, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?
2. Emnggunakan baris kode Buku14 bk1 = new Buku14();
3. Bagaimana cara mengakses atribut dan method dari suatu objek?

menggunakan sintaks namaObjek.namaAtribut untuk mengakses atribut dan namaObjek.namaMethod() untuk memanggil method. Misalnya:





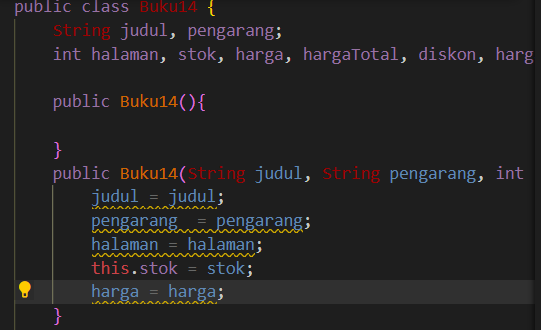
1. Mengapa hasil output pemanggilan method tampilInformasi() pertama dan kedua berbeda?

adanya perubahan nilai pada atribut-atribut objek di antara dua pemanggilan tersebut. Misalnya, jika terdapat pemanggilan terjual(5) dan gantiHarga(60000) di antara dua pemanggilan tampilInformasi(), maka informasi yang ditampilkan pada kedua pemanggilan tampilInformasi() akan berbeda karena stok dan harga buku telah berubah

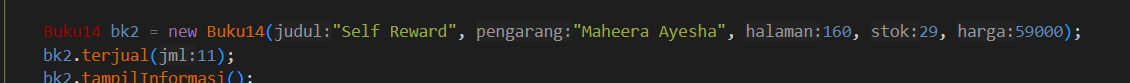
**Percobaan 3: Membuat Konstruktor**

Buat class buku, Tambahkan dua buah konstruktor di dalam class Buku tersebut, yang

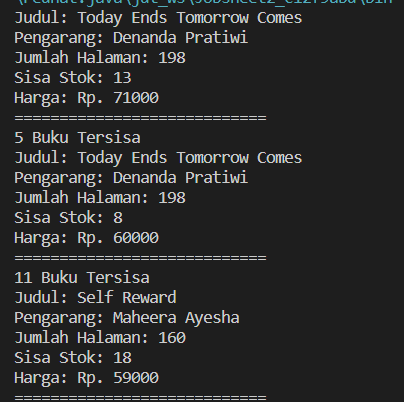
terdiri dari satu konstruktor default dan satu konstruktor berparameter.



Buka kembali class BukuMain. Buat sebuah object lagi bernama bk2 dengan menggunakan konstruktor berparameter

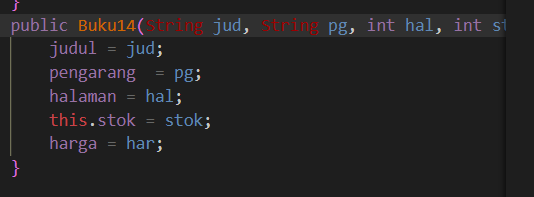


Maka dapat dihasilkan



**Pertanyaan**

* + - 1. Pada class **Buku** di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!



* + - 1. Perhatikan class **BukuMain**. Apa sebenarnya yang dilakukan pada baris program berikut?



Pada baris program tersebut, objek bk2 dari kelas Buku14 dibuat dengan menggunakan konstruktor berparameter. Parameter-parameter yang diberikan adalah judul buku ("Self Reward"), nama pengarang ("Maheera Ayesha"), jumlah halaman (160), jumlah stok (29), dan harga (59000).

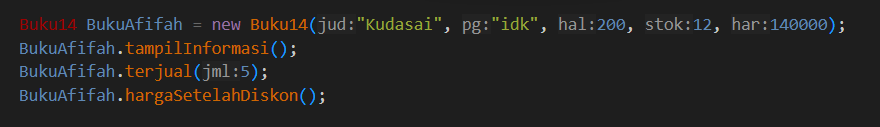
* + - 1. Hapus konstruktor default pada class **Buku**, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!

muncul kesalahan (error) yang menyatakan bahwa tidak ada konstruktor default yang tersedia. Hal ini terjadi karena ketika tidak ada konstruktor default yang tersedia, Java tidak dapat membuat instance dari kelas tersebut secara implisit. Untuk mengatasi hal ini,perlu menyediakan konstruktor yang memungkinkan inisialisasi objek tanpa harus menggunakan konstruktor default.

* + - 1. Setelah melakukan instansiasi object, apakah method di dalam class **Buku** harus diakses secara berurutan? Jelaskan alasannya!

Buku tidak harus diakses secara berurutan karena dapat memanggil method sesuai dengan kebutuhan dan urutan yang diinginkan. Hal ini karena setiap method di dalam class Buku tidak bergantung satu sama lainnya, dan dapat dipanggil secara independen.

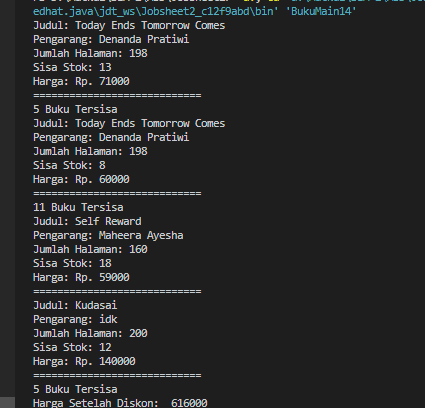
* + - 1. Buat object baru dengan nama **buku<NamaMahasiswa>** menggunakan konstruktor berparameter dari class **Buku**!



**Latihan Praktikum**

1. 

Hasil program



1. 

Hasil Program

