# LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

Dosen Pengajar : Triana Fatmawati, S.T, M.T

# Jobsheet-2 Object



Nama : Surya Rahmat Fatahillah

NIM : 2341760020

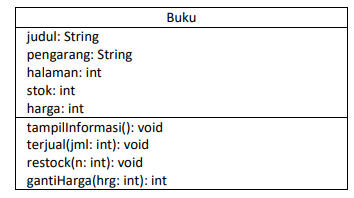
Prodi : Sistem Informasi Bisnis

# JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG 2024

**2. Praktikum**

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

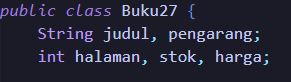
Pada Percobaan 1 ini dilakukan pembuatan class beserta atribut dan method yang dimilikinya. Perhatikan Class Diagram berikut ini:

****

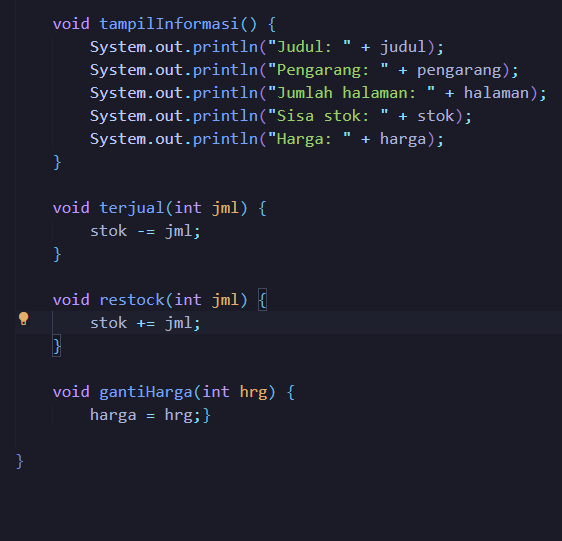
Berdasarkan class diagram tersebut, akan dibuat program mengguanakan bahasa Java.

**2.1.1 Langkah-langkah Percobaan**

1. Buka text editor. Buat file baru, beri nama Buku27.java
2. Lengkapi class Buku dengan atribut yang telah digambarkan di dalam class diagram tersebut.

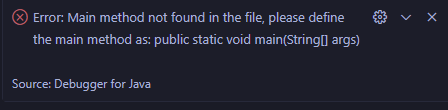
****

1. Lengkapi class Buku dengan method yang telah digambarkan di dalam class diagram tersebut.

****

1. Compile dan run program.

**2.1.2 Verifikasi Hasil Percobaan**

****

Kode tersebut mengalami error karena tidak ada main method nya

**2.1.3 Pertanyaan**

1. Sebutkan dua karakteristik class atau object!

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

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

4. 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)!

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

6. Commit dan push kode program ke Github

**Jawaban**

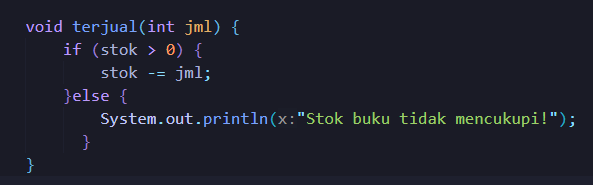
## 1. Dua karakteristik utama class atau object:

1. **Atribut (State):** Merupakan data atau informasi yang dimiliki oleh class atau object. Atribut ini mendefinisikan keadaan atau properti dari class atau object tersebut. Contohnya, atribut pada class "Mobil" dapat berupa merk, warna, tahun produksi, dan jumlah kursi.
2. **Behavior atau Method:** Merupakan tindakan atau operasi yang dapat dilakukan oleh class atau object. Behavior ini mendefinisikan apa yang dapat dilakukan oleh class atau object tersebut. Contohnya, behavior pada class "Mobil" dapat berupa berjalan, berhenti, berbelok, dan membuka pintu.

2. Pada class Buku terdapat 5 attribut yaitu Judul, pengarang, halaman, stok, dan harga

3. pada class Buku terdapat 4 method yaitu tampilInformasi(): void, terjual(jml: int): void, restock(n: int): void, gantiHarga(hrg: int): int

4. Berikut modifikasi isi method terjual() sehingga proses pengurangan hanya dapat dilakukan jika stok masih ada (lebih besar dari 0)!





5. Method restock() mempunyai satu parameter berupa bilangan int karena method tersebut digunakan untuk menambah jumlah stok buku. Parameter int ini mewakili jumlah buku yang ingin ditambahkan ke stok. pengguna dapat memasukan jumlah buku yang akan ditambahkan ke stok ketika mereka memanggil method restock. Misalnya, jika seseorang ingin menambahkan 5 buku ke stok, mereka dapat menggunakan pernyataan buku1.restock(5);

6. Commit dan push program ke Github:

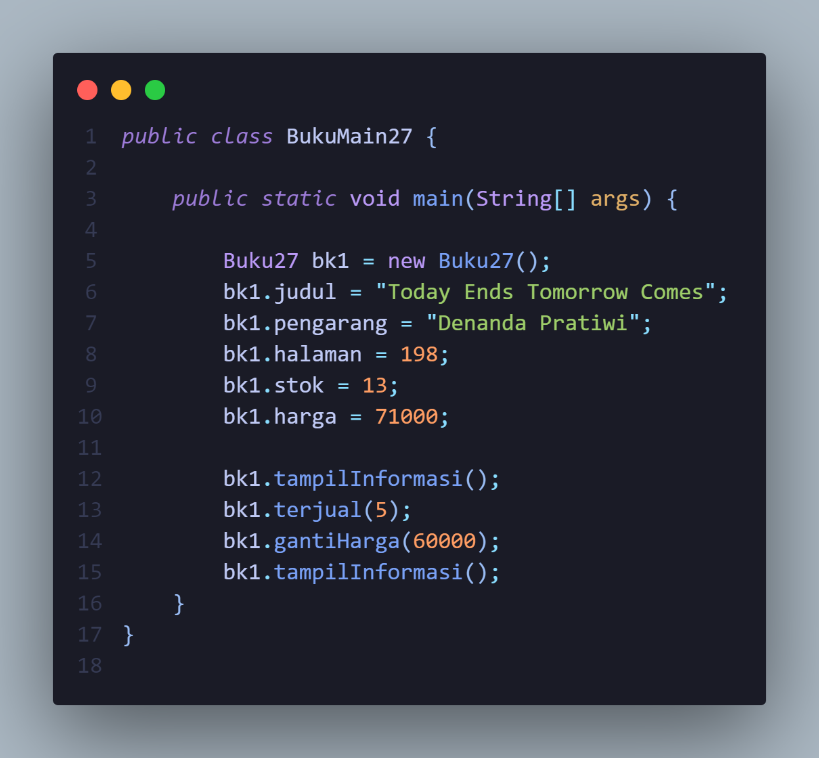
**2.2 Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method**

**2.2.1 Langkah-langkah Percobaan**

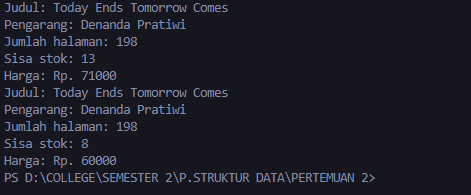
1. Buat file baru, beri nama BukuMain27.java

2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi main()

3. Di dalam fungsi main(), lakukan instansiasi, kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.

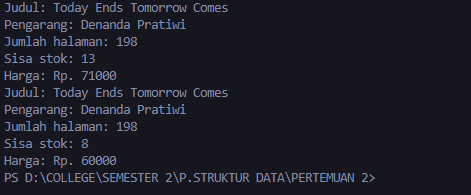


4. Compile dan run Program



5. Commit dan push kode program ke Github

**2.2.2 Verifikasi Hasil Percobaan**

****

**2.2.3 Pertanyaan**

1. Pada class BukuMain, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?

2. Bagaimana cara mengakses atribut dan method dari suatu objek?

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

**Jawaban!**

1. Pada class Buku main berikut merupakan baris kode program yang digunakan untuk proses instansiasi



Dan nama object yang dihasilkan yaitu bk1

1. Yang pertama yaitu harus menginstansiasi objek dari suatu kelas, kemudian setelah membuat object maka selanjutnya yaitu,gunakan nama objek diikuti dengan tanda titik (.) dan kemudian nama atribut atau method yang ingin diakses.

Contoh:

* 1. Mengakses atribut judul: buku1.judul
  2. Mengakses method tampilInformasi: buku1.tampilInformasi()

1. Hasil output pemanggilan method tampilInformasi() pertama dan kedua berbeda karena nilai atribut stok dan harga buku telah diubah setelah pemanggilan pertama. Pada pemanggilan pertama, method tampilInformasi() mencetak nilai awal atribut stok dan harga yang diinisialisasi pada saat pembuatan objek bk1. Setelah pemanggilan pertama, method terjual(5) mengurangi nilai stok buku sebanyak 5. Method gantiHarga(60000) mengubah nilai harga buku menjadi 60.000. Pada pemanggilan kedua, method tampilInformasi() mencetak nilai terbaru dari atribut stok dan harga yang telah diubah.

**2.3 Percobaan 3: Membuat Konstruktor**

**2.3.1 Langkah-langkah Percobaan**

1. Buka kembali class Buku27. Tambahkan dua buah konstruktor di dalam class Buku tersebut, yang terdiri dari satu konstruktor default dan satu konstruktor berparameter. Konstruktor merupakan method Istimewa, penempatan kode program untuk konstruktor dapat diperlakukan sama seperti method yang lain (setelah atribut).



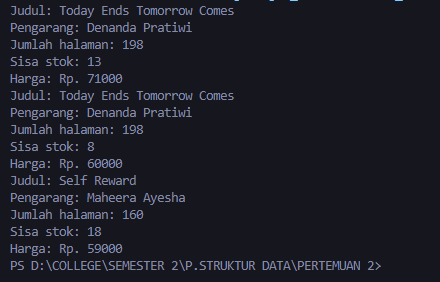
2. Buka kembali class BukuMain. Buat sebuah object lagi bernama bk2 dengan menggunakan konstruktor berparameter



3. Compile dan run program.

4. Commit dan push kode program ke Github

**2.3.2 Verifikasi Hasil Percobaan**

****

**2.3.3 Pertanyaan**

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

2. Perhatikan class BukuMain. Apa sebenarnya yang dilakukan pada baris program berikut?



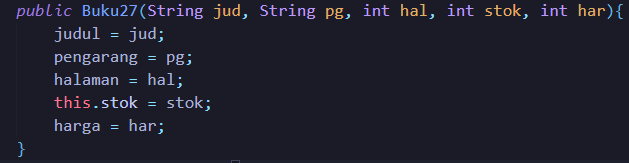
3. Hapus konstruktor default pada class Buku, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian

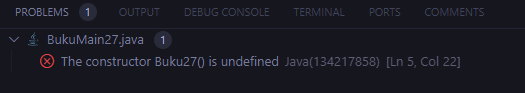
4. Setelah melakukan instansiasi object, apakah method di dalam class Buku harus diakses secara berurutan? Jelaskan alasannya!

5. Buat object baru dengan nama buku<NamaMahasiswa> menggunakan konstruktor berparameter dari class Buku!

6. Commit dan push kode program ke Github

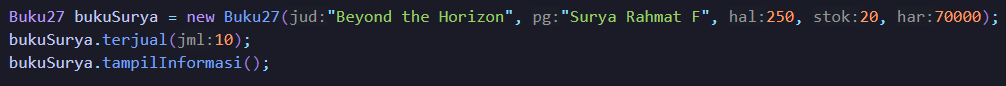
**Jawaban!**

1. Berikut merupakan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter
2. Baris program Buku27 bk2 = new Buku27("Self Reward", "Maheera Ayesha", 160, 29, 59000); yang sebenarnya dilakukan adalah menciptakan sebuah objek baru bernama bk2 dari class Buku27 dengan menginisialisasi nilai awal untuk atribut judul sebagai "Self Reward", pengarang sebagai "Maheera Ayesha", halaman sebanyak 160, stok sebanyak 29, dan harga senilai 59000.
3. Ketika konstruktor default dihapus pada class Buku27, program tidak akan compile dan menghasilkan error berikut:



Hal ini dapat terjadi karena, Saat konstruktor default dihapus, class Buku27 tidak memiliki konstruktor yang dapat digunakan untuk membuat objek baru tanpa parameter. Baris program Buku27 bk1 = new Buku27(); merupakan program yang mencoba membuat objek baru dari class Buku27 tanpa parameter. Karena tidak ada konstruktor yang sesuai, program tidak dapat compile dan error.

1. Tidak, method di dalam class Buku tidak harus diakses secara berurutan. Pengaksesan method dapat dilakukan secara bebas tanpa memperhatikan urutan tertentu. Hal ini disebabkan karena setiap method di dalam class Buku bersifat independen dan dapat dipanggil sesuai kebutuhan tanpa ketergantungan pada urutan eksekusi yang lain.
2. Berikut merupakan kode program untuk pembuatan object Bernama buku dengan menggunakan konstruktor berparameter dari class Buku:

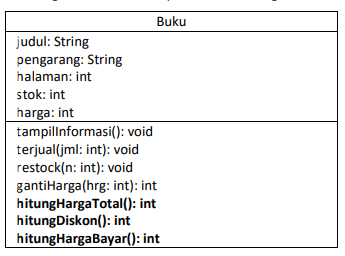


1. Commit dan push ke Github

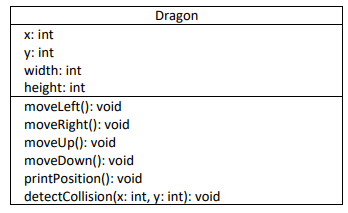
**2.4 Latihan Praktikum**

1. Pada class Buku yang telah dibuat, tambahkan tiga method yaitu hitungHargaTotal(), hitungDiskon(), dan hitungHargaBayar() dengan penjelasan sebagai berikut: o Method hitungHargaTotal() digunakan untuk menghitung harga total yang merupakan perkalian antara harga dengan jumlah buku yang terjual o Method hitungDiskon() digunakan untuk menghitung diskon dengan aturan berikut: ▪ Jika harga total lebih dari 150000, maka harga didiskon sebesar 12% ▪ Jika harga total antara 75000 sampai 150000, maka harga didiskon sebesar 5% ▪ Jika harga total kurang dari 75000, maka harga tidak didiskon o Method hitungHargaBayar() digunakan untuk menghitung harga total setelah dikurangi diskon

Class diagram Buku setelah penambahan ketiga method tersebut adalah sebagai berikut.



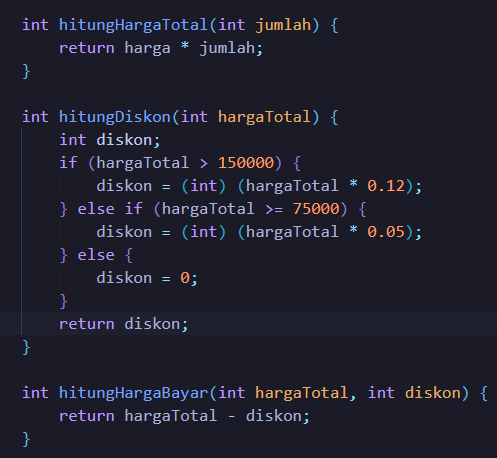
2. Buat program berdasarkan class diagram berikut ini!



Penjelasan dari atribut dan method pada class Dragon tersebut adalah sebagai berikut: • Atribut x digunakan untuk menyimpan posisi koordinat x (mendatar) dari dragon, sedangkan atribut y untuk posisi koordinat y (vertikal) • Atribut width digunakan untuk menyimpan lebar dari area permainan, sedangkan height untuk menyimpan panjang area • Method moveLeft() digunakan untuk mengubah posisi dragon ke kiri (koordinat x akan berkurang 1), sedangkan moveRight() untuk bergerak ke kanan (koordinat x akan bertambah 1). Perlu diperhatikan bahwa koordinat x tidak boleh lebih kecil dari 0 atau lebih besar dari nilai width. Jika koordinat x < 0 atau x > width maka panggil method detectCollision() • Method moveUp() digunakan untuk mengubah posisi dragon ke atas (koordinat y akan berkurang 1), sedangkan moveDown() untuk bergerak ke bawah (koordinat y akan bertambah 1). Perlu diperhatikan bahwa koordinat y tidak boleh lebih kecil dari 0 atau lebih besar dari nilai height. Jika koordinat y < 0 atau y > height maka panggil method detectCollision() • Method detectCollision() akan mencetak pesan “Game Over” apabila dragon menyentuh ujung area permainan

**Jawaban!**

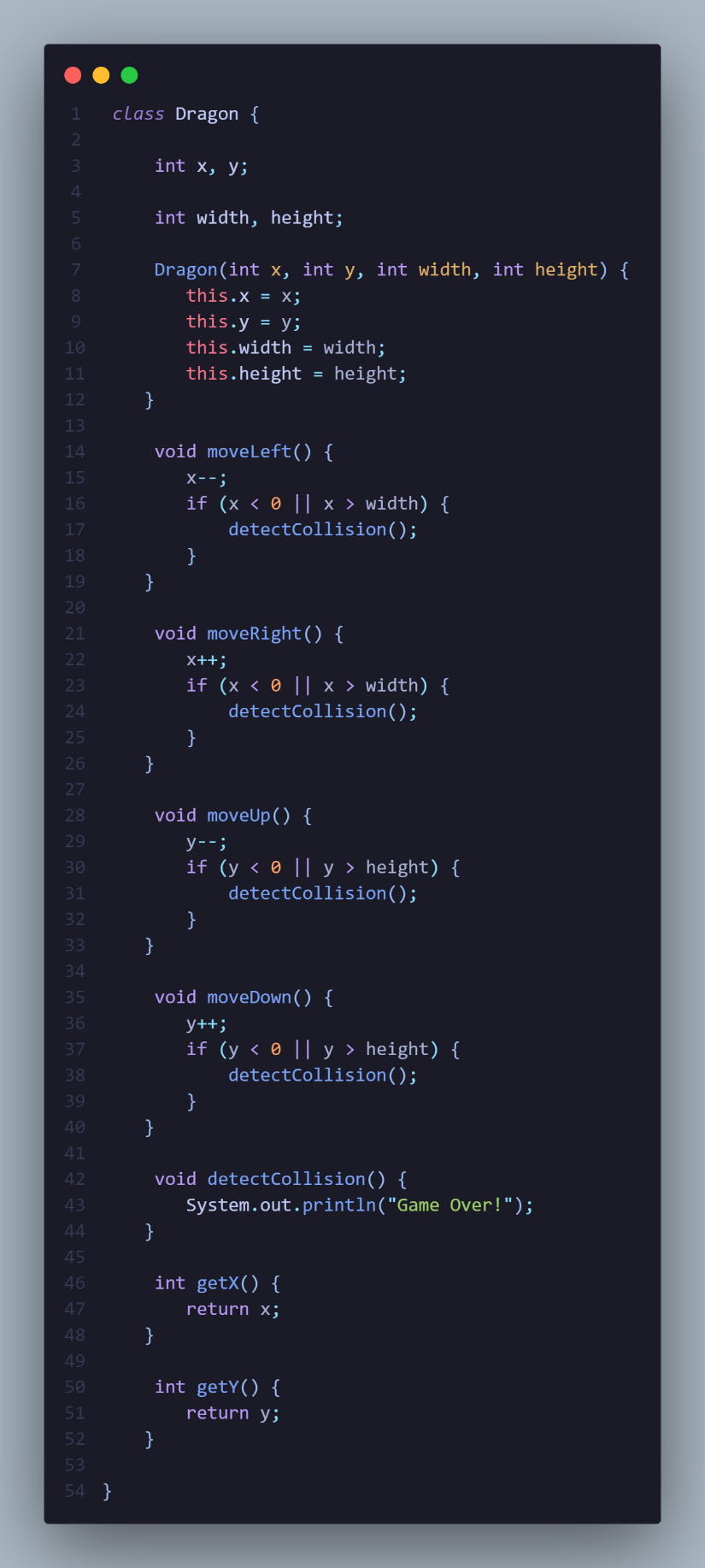
1. Berikut class buku yang telah ditambahkan 3 method yaitu hitungHargaTotal(), hitungDiskon(), dan hitungHargaBayar().





1. Berikut merupakan program berdasarkan class Dragon pada diagaram!

* Class Dragon



* Class Dragon Main
* Ouput

