**LAPORAN RESMI**

**MODUL III**

**INHERINTANCE & OVERRIDE**

**PEMROGRAMAN BERBASIS OBJEK**

**Disetujui : 27 MEI 2024**

**Asisten**

**MUHAMMAD ZAIN MAULANA**

**20.04.411.00019**

**NAMA : ABAS PERMADANI**

**N.R.P : 230441100107**

**DOSEN : ACHMAD ZAIN NUR, S.KOM., M.T.**

**ASISTEN : MUHAMMAD ROSYID MAULANA**

**TGL PRAKTIKUM : 03 MEI 2024**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Dalam pemrograman berorientasi objek (OOP), "Inheritance" (Pewarisan) dan "Override" (Penimpaan) merupakan dua konsep kunci yang memungkinkan pemrogram untuk membuat software yang lebih modular, efisien, dan mudah untuk dikelola

Inheritance (Pewarisan) Reuse Kode: Pewarisan memungkinkan developers untuk menggunakan kembali kode yang sudah ada. Sebagai contoh, jika Anda memiliki class yang mendefinisikan karakteristik dasar dari semua kendaraan, Anda bisa mewariskan class ini untuk membuat class yang lebih spesifik seperti mobil atau sepeda motor tanpa harus menulis ulang semua kode yang sama.Strukturisasi Kode: Dengan pewarisan, kita bisa mengatur kode dalam hierarki yang jelas, di mana class dasar menyediakan fitur dasar, dan class turunan bisa menambah atau mengkustomisasi fitur tersebut. Hal ini memudahkan dalam pemahaman dan pengelolaan kode.Abstraksi: Pewarisan memungkinkan pemrogram untuk bekerja pada tingkat abstraksi yang lebih tinggi. Developer bisa menangani objek pada tingkat umum dalam hierarki, dan membiarkan detail implementasi pada subclass.

## **Tujuan**

* Mahasiswa mampu memahami konsep Inherintance atau pewarisan dalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya.
* Mahasiswa mampu memahami konsep override dalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya.

# **BAB II DASAR TEORI**

## **Pengertian inheritance :**

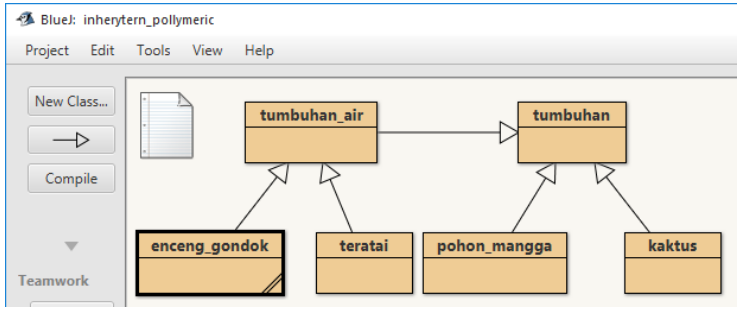
Inheritance adalah salah satu konsep dasar dalam java. Inheritance memiliki arti yaitu warisan jadi dalam java ada class yang akan mewariskan method atau fungsi dan ada yang akan menerima warisan itu. Pemberi warisan disebut superclass sedangkan untuk penerimanya disebut subclass, atau biasa juga disebut sebagai induk dan anak class.



## **Pengertian override :**

Override memiliki artian yaitu menimpa, dalam java method superclass dapat ditimpa oleh subclass dengan cara menuliskan lagi nama method yang sama pada subclass. Override biasanya dilakukan jika method superclassnya tidak memiliki makna yang sama atau isi dari method superclass harus dirubah.

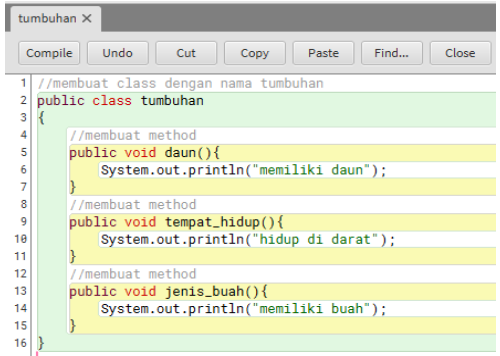
**Contoh program :**

****

Class program :

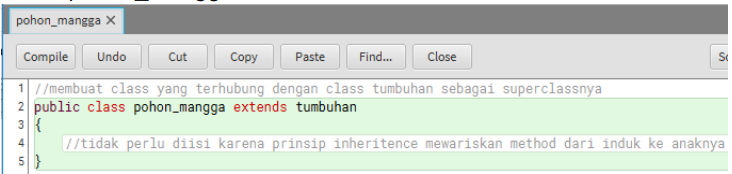
* Class tumbuhan sebagai superclass semua method awal dituliskan disini
* Class tumbuhan memiliki subclass yaitu pohon\_mangga, dan kaktus.
* Tumbuhan air memiliki subclass yaitu enceng\_gondok, dan teratai

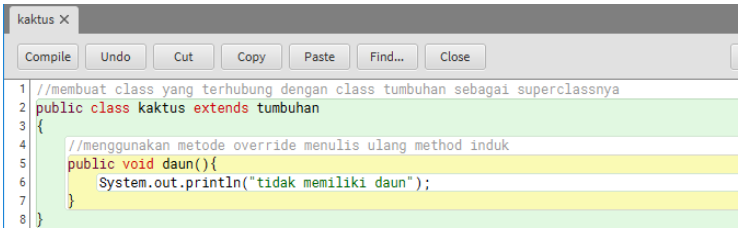
Class tumbuhan :

****

Class tumbuhan membuat 3 method yang nantinya akan diturunkan ke setiap subclassnya.

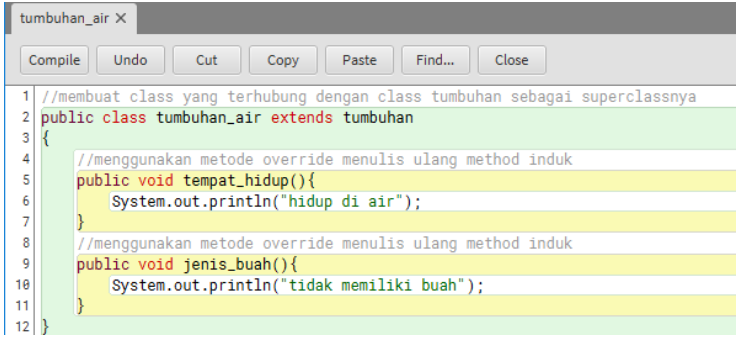
Class pohon\_mangga dan kaktus :

****

****

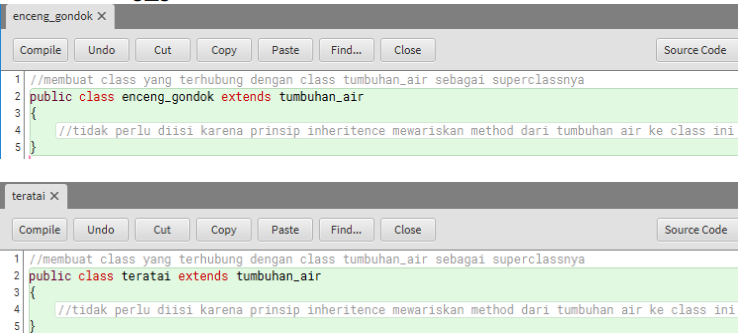
Class pohon\_mangga menerima semua warisan dari class tumbuhan, sedangkan kaktus menimpa (override) method daun karena method daun pada class tumbuhan tidak memiliki makna yang sama dan harus diganti.

Class tumbuhan\_air :

****

Class tumbuhan\_air menimpa class tumbuhan pada method tempat hidup dan jenis\_buah.

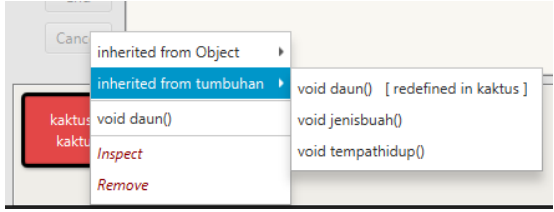
Class enceng\_gondok dan teratai :

****

Class enceng\_gondok dan teratai terhubung dengan class tumbuhan air sehingga sudah tidak terpengaruh dengan clas tumbuhan lagi, jadi method yang digunakan adalah method yang ditimpa di class tumbuhan\_air dan method warisan yang tidak ditulis ulang.

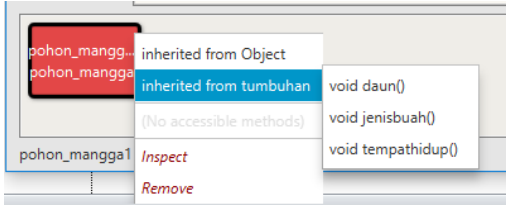
**Running program :**

Class kaktus :



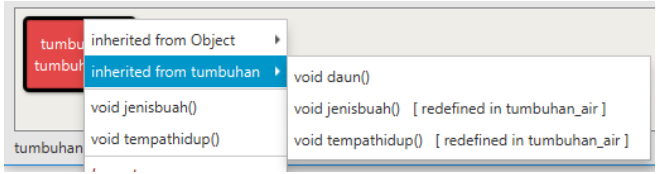
Perhatikan pada inherited from tumbuhan jadi diturunkan dari tumbuhan, pada method daun ada keterangan redefined in kaktus, jadi method daun telah ditimpa atau ditulis ulang pada class kaktus.

Class pohon\_mangga :



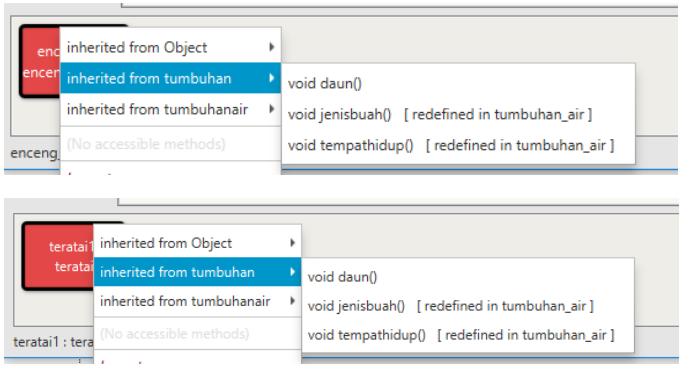
Pada inherited from tumbuhan pada semua methodnya tidak ada redefined berarti class ini mewarisi semua yang ada di classs tumbuhan.

Class tumbuhan\_air :

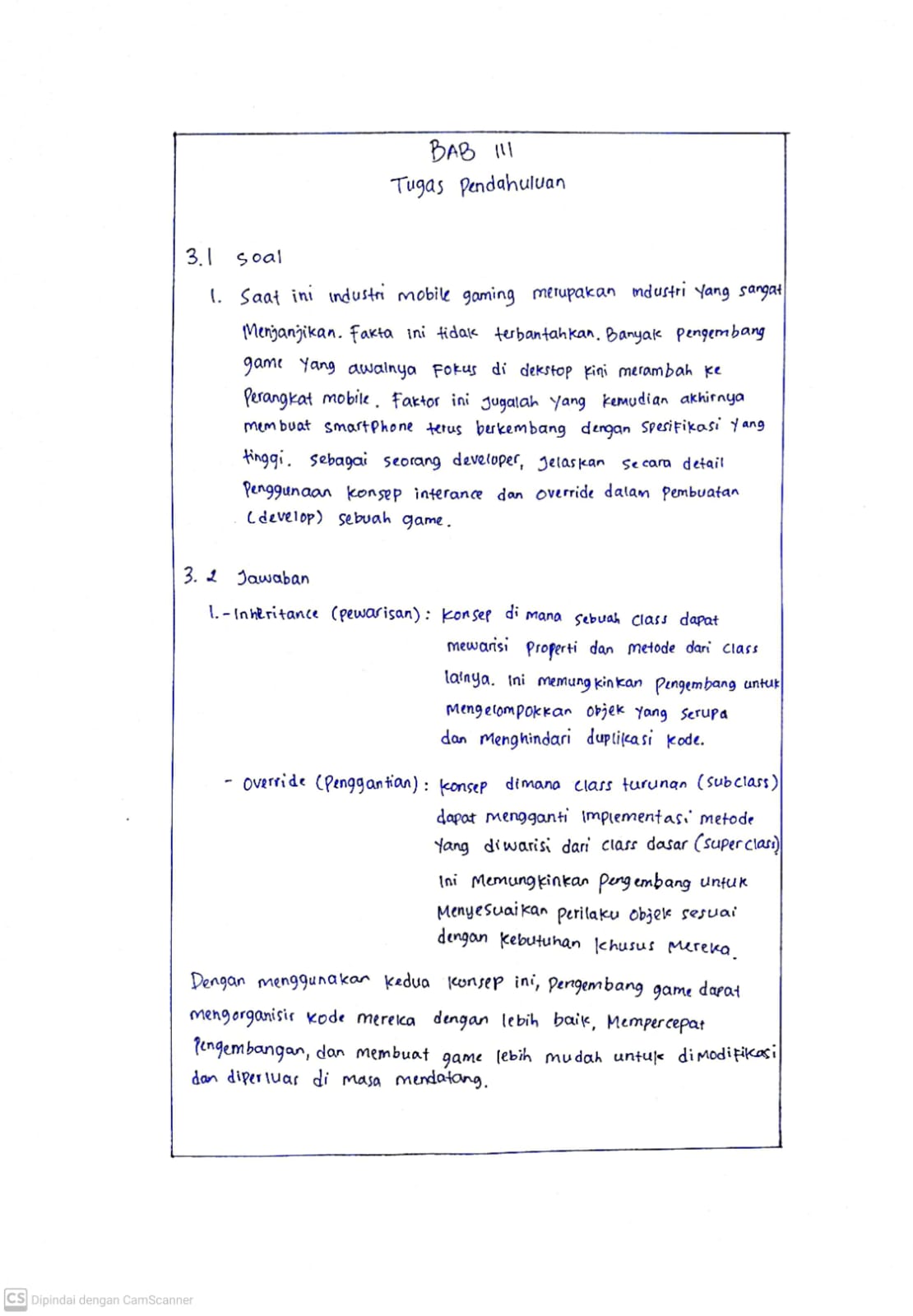


Class tumbuhan\_air menimpa atau menulis ulang class tumbuhan pada bagian mehod jenisbuah dan tempathidup.

Class enceng\_gondok dan teratai :



Pada class enceng\_gondok dan teratai dapat dilihat mereka menerima method dari class tumbuhan air karena pada method jenisbuah dan tempathidup telah ditulis ulang oleh class tumbuhan air sedangkan untuk daun masih warisan dari class tumbuhan.

****

# **BAB IV IMPLEMENTASI**

## **Tugas praktikum**

Buatlah aplikasi perpustakaan sederhana sebagai berikut: Masukkan data buku sebanyak X kali

1. Judul : Judul buku.

2. Penulis : Nama penulis film tersebut.

3. Publisher : Perusahaan yang menerbitkan buku tersebut.

4. Kategori : SU = Semua Umur, D = Dewasa, R = Remaja, A = Anak-anak

5. Stok : Jumlah stok buku tersebut.

6. Tahun Terbit : Tahun buku tersebut diterbitkan

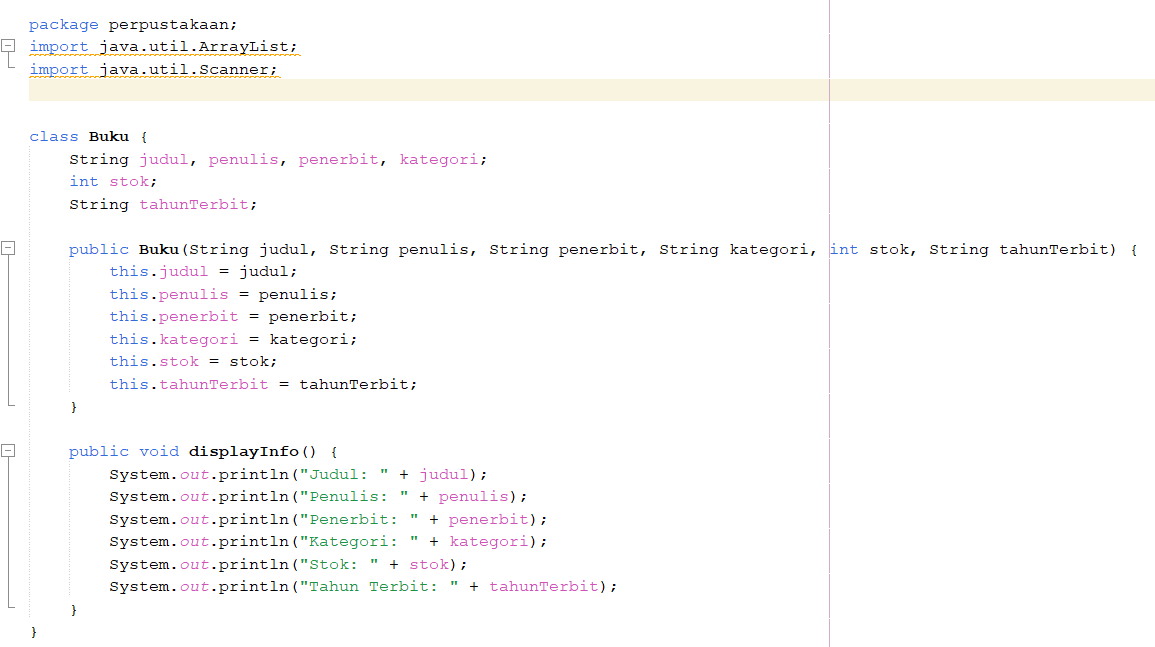
Judul, Penulis, Publisher, Kategori, Stok, dan Tahun Terbit di-looping

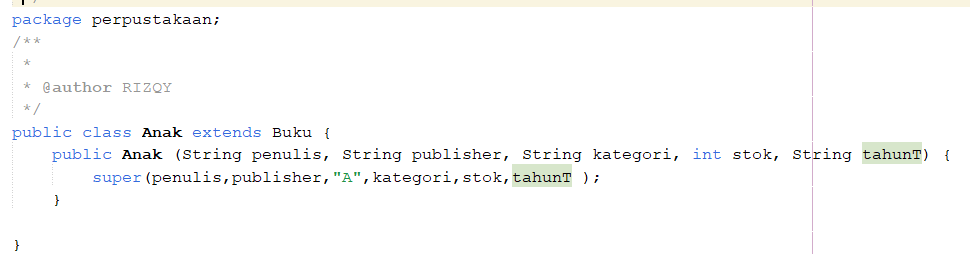
sebanyak X kali.

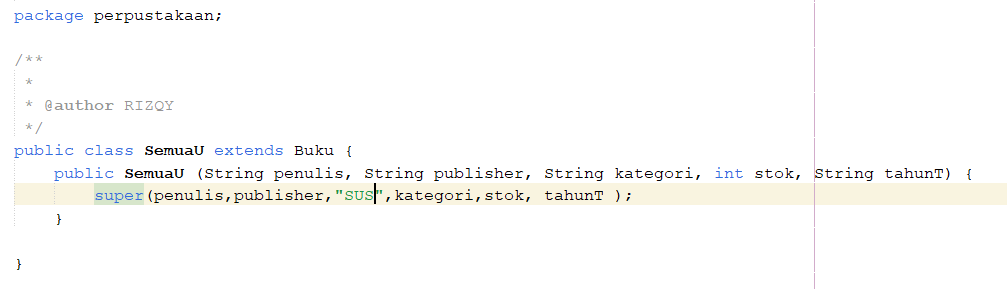
Desainlah aplikasi perpustakaan tersebut dengan konsep inheritance dan tentukan parent class serta child class-nya. Setelah itu, implementasikan class-class yang telah didesain dengan membuat program sederhana yang memiliki fasilitas entri data buku dan melihat daftar buku yang telahdi-entri-

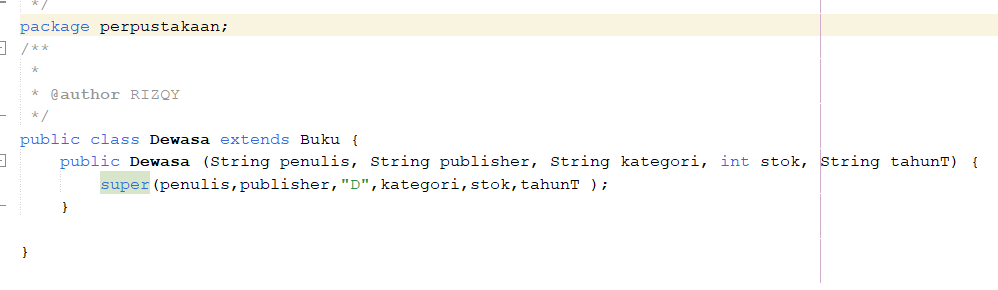
kan.

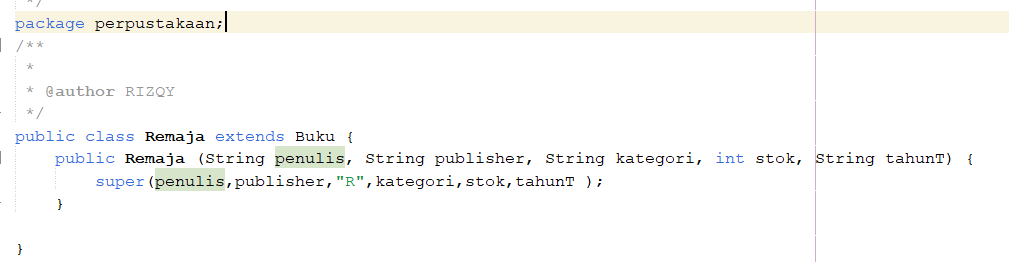
## **Source code**

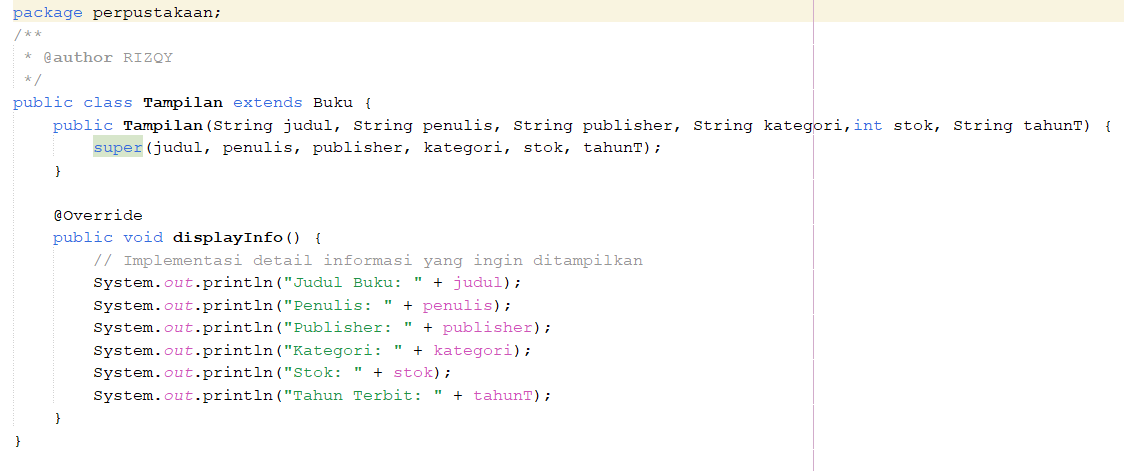


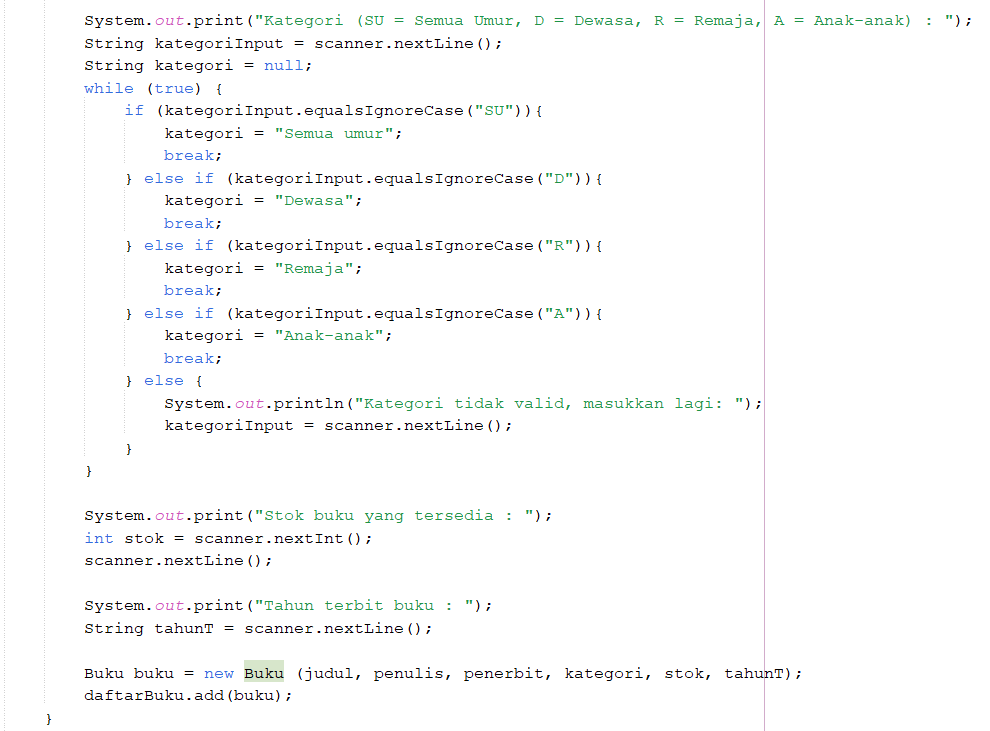


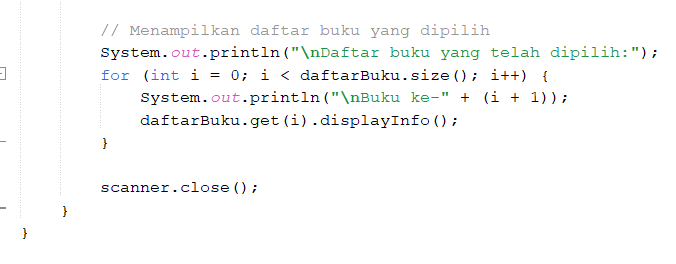




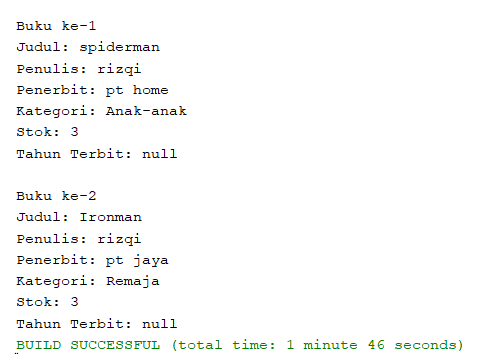








## **Hasil output**



## **4.4 Penjelasan**

Disini saya membuat program beberapa buku, yang terdiri dari judul, penulis, penerbit, kategori, stok, tahun terbit . Mengulangi entri data untuk setiap buku. Di dalam lingkaran : Input Detail Buku : Ini meminta pengguna untuk setiap judul buku, penulis, penerbit, kategori, stok, dan tahun penerbitan. Validasi Kategori : Memastikan kategori input valid (SU, D, R, A). Jika tidak, ia bertanya lagi. Membuat dan Menyimpan Objek Buku : Membuat Bukuobjek dengan detail yang disediakan dan menambahkannya ke daftarBukudaftar. Program ini dirancang untuk mengelola perpustakaan buku dengan memungkinkan pengguna memasukkan detail beberapa buku dan kemudian menampilkan detail tersebut. Ini menggunakan prinsip pemrograman berorientasi objek dengan merangkum detail buku di Bukukelas dan mengelola banyak buku menggunakan file ArrayList. Program ini menangani masukan pengguna dan memvalidasinya, khususnya untuk kategori buku, memastikan entri data yang kuat.

# **BAB V PENUTUP**

## **Analisa**

Inheritance(Pewarisan)Definisi: Inheritance adalah sebuah mekanisme di dalam pemrograman berorientasi objek yang memungkinkan sebuah class untuk mewarisi atribut dan metode dari class lain. Class yang mewarisi disebut sebagai subclass (atau derived class, child class), sedangkan class yang diwarisi disebut superclass (atau base class, parent class).Keuntungan : Pengurangan Redundansi: Pewarisan memungkinkan penggunaan kembali kode yang sudah ada tanpa harus menulis ulang. Ini mengurangi redundansi kode dan memperbaiki efisiensi pengembangan. Pemeliharaan yang Mudah: Perubahan pada superclass akan otomatis terdistribusi ke subclass, kecuali jika fungsi tersebut di-override. Ini memudahkan pemeliharaan dan pengelolaan kode. Kekurangan : Ketergantungan Berlebihan: Terlalu banyak pewarisan dapat membuat struktur class menjadi rumit dan sulit untuk ditelusuri.

Override (Penimpaan) Definisi : Override adalah mekanisme di mana subclass memodifikasi implementasi dari metode yang diwarisi dari superclass. Subclass menggunakan nama yang sama untuk metode tersebut, tetapi mengubah perilaku yang didefinisikan dalam superclass. Keuntungan : Kustomisasi Perilaku: Override memungkinkan subclass untuk memodifikasi atau memperluas perilaku yang diwarisi dari superclass sesuai dengan kebutuhan yang lebih spesifik. Polimorfisme: Override adalah kunci dari polimorfisme, di mana metode yang sama dapat memiliki perilaku yang berbeda berdasarkan objek mana yang memanggil metode tersebut. Kekurangan : Kompleksitas Kode: Jika banyak metode di-override, hal ini bisa membuat kode sulit untuk dipahami dan dipelihara. Resiko Tersembunyi: Jika tidak dirancang dengan hati-hati, override dapat menyebabkan perilaku yang tidak diinginkan karena mengubah fungsi yang mungkin tidak sepenuhnya dipahami oleh pengembang subclass.

## **Kesimpulan**

**Inheritance (Pewarisan)**

* Pengurangan Redundansi: Mengizinkan penggunaan kembali kode, mengurangi kebutuhan untuk menulis ulang kode yang sama.
* Pemeliharaan yang Mudah: Perubahan pada superclass secara otomatis terdistribusi ke subclass, memudahkan pemeliharaan.
* Polimorfisme: Memungkinkan objek dari subclass diperlakukan sebagai objek dari superclass, memberikan fleksibilitas dalam pemanggilan metode.
* Ketergantungan Berlebihan: Bisa membuat struktur class rumit dan sulit untuk ditelusuri jika digunakan secara berlebihan.
* Kopling yang Kuat: Subclass yang terlalu bergantung pada perilaku superclass dapat menjadi masalah jika terjadi perubahan besar pada superclass.

**Override (Penimpaan)**

* Kustomisasi Perilaku: Memungkinkan subclass untuk mengubah atau memperluas perilaku yang diwarisi dari superclass.
* Polimorfisme: Kunci dari polimorfisme, dimana metode yang sama bisa memiliki perilaku yang berbeda tergantung objek yang memanggilnya.
* Kompleksitas Kode: Membuat kode menjadi lebih sulit untuk dipahami jika terlalu banyak metode yang di-override.
* Resiko Tersembunyi: Bisa menyebabkan perilaku yang tidak diinginkan atau bug jika override tidak dilakukan dengan hati-hati.