**Tugas Akhir**

**Sistem Manajemen Inventaris Tanaman Hias Cerdas (Smart Plant Inventory Management - SPIM)**

**Versi Dual Mode: Console & GUI**

Disusun Untuk Memenuhi Tugas Pemrograman Berorientasi Objek

Dosen Pengampu : Ali Tarmuji, S.T., M.Cs.



**Disusun oleh:**

**NAMA : Anggun Septiani**

**NIM : 2400018147**

**KELAS : C**

**Link Github**

**<https://github.com/Anggunseptiani/TUGAS-AKHIR_PBO2025>**

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS AHMAD DAHLAN**

**2025/2026**

# **BAB I: PENDAHULUAN**

## **1.1 Latar Belakang**

# Perkembangan teknologi informasi pada era digital saat ini telah memberikan dampak besar terhadap berbagai sektor usaha, termasuk usaha perdagangan tanaman hias. Tanaman hias merupakan salah satu komoditas hortikultura yang memiliki nilai ekonomis tinggi dan semakin diminati masyarakat Indonesia, baik sebagai hobi, dekorasi rumah, maupun sebagai peluang bisnis yang menjanjikan. Tingginya minat masyarakat terhadap tanaman hias menyebabkan meningkatnya kebutuhan akan pengelolaan inventaris yang efektif dan efisien pada toko tanaman, nursery, dan kebun pribadi skala menengah.

# Namun, pada kenyataannya masih banyak pelaku usaha tanaman hias skala kecil hingga menengah yang melakukan pencatatan data inventaris secara manual, seperti menggunakan buku tulis atau catatan sederhana. Cara ini memiliki berbagai kelemahan, antara lain risiko kesalahan pencatatan, data yang mudah hilang, sulitnya melakukan monitoring kondisi tanaman secara berkala, kesulitan dalam tracking stok yang tersedia, serta kurang efisien dalam jangka panjang.

# Dalam praktiknya, pengelolaan inventaris tanaman secara manual seringkali menghadapi berbagai kendala, seperti:

# Kesulitan dalam tracking jumlah stok yang tersedia

# Sulitnya monitoring kondisi kesehatan tanaman secara berkala

# Pencatatan data yang tidak terstruktur dan tidak konsisten

# Kesulitan dalam membuat laporan inventaris dan statistik

# Risiko kehilangan data penting akibat kerusakan catatan manual

# Ketidakakuratan dalam pencatatan karakteristik khusus tanaman

# Seiring dengan perkembangan teknologi, pemanfaatan aplikasi komputer menjadi solusi yang tepat untuk mengatasi permasalahan tersebut. Dengan adanya aplikasi pengelolaan inventaris tanaman hias berbasis teknologi informasi, proses pencatatan data tanaman, monitoring kondisi, tracking stok, hingga pembuatan laporan statistik dapat dilakukan secara otomatis, cepat, akurat, dan terstruktur.

# Berdasarkan hal tersebut, penulis merancang sebuah aplikasi berbasis Java yang diberi judul **SPIM (Sistem Pengelolaan Inventaris Tanaman Hias)**. Aplikasi ini hadir dalam **dua mode**: **Mode Console** untuk pengguna yang menginginkan antarmuka berbasis teks yang efisien, dan **Mode GUI (Graphical User Interface)** untuk pengguna yang menginginkan pengalaman visual yang lebih modern dan interaktif. Kedua mode ini menggunakan sistem penyimpanan data yang sama, sehingga data dapat diakses dan dikelola dari kedua antarmuka.

# Aplikasi ini dirancang dengan menerapkan konsep Object Oriented Programming (OOP), yaitu enkapsulasi, pewarisan, dan polimorfisme secara nyata dan terstruktur. Aplikasi ini diharapkan dapat menjadi media pembelajaran sekaligus contoh penerapan OOP dalam kasus nyata yang memiliki nilai guna praktis.

# Penggunaan teknologi informasi dalam bentuk aplikasi manajemen inventaris berbasis Object-Oriented Programming (OOP) dapat menjadi solusi komprehensif untuk permasalahan tersebut. Dengan menerapkan konsep OOP yang meliputi enkapsulasi, pewarisan, dan polimorfisme, sistem dapat dirancang dengan struktur yang lebih terorganisir, mudah dikembangkan, maintainable, dan sesuai dengan prinsip pemrograman yang baik.

## **1.2 Rumusan Masalah**

# Adapun rumusan masalah dalam pembuatan aplikasi ini adalah sebagai berikut:

# Bagaimana merancang aplikasi pengelolaan inventaris tanaman hias berbasis Java yang dapat mengelola data tanaman secara efektif dan efisien?

# Bagaimana menerapkan konsep Object Oriented Programming (enkapsulasi, pewarisan, dan polimorfisme) pada aplikasi SPIM?

# Bagaimana membuat dual interface (Console dan GUI) yang dapat digunakan secara bergantian dengan data yang konsisten?

# Bagaimana mengimplementasikan fitur CRUD (Create, Read, Update, Delete) untuk pengelolaan data tanaman dalam sistem?

# Bagaimana melakukan penyimpanan data secara persisten menggunakan file teks agar data tidak hilang saat aplikasi ditutup?

# Bagaimana membuat sistem pelaporan statistik yang dapat memberikan informasi kondisi dan stok tanaman secara otomatis?

# Bagaimana mengimplementasikan GUI yang user-friendly dengan Java Swing yang terintegrasi dengan sistem backend?

## **1.3 Tujuan Pembuatan Aplikasi**

# Tujuan dari pembuatan aplikasi SPIM adalah:

# Merancang dan mengimplementasikan aplikasi pengelolaan inventaris tanaman hias berbasis Java dengan struktur program yang baik

# Menerapkan konsep Object Oriented Programming secara nyata dalam pembuatan aplikasi, meliputi:

# Enkapsulasi (Encapsulation) untuk melindungi data

# Pewarisan (Inheritance) untuk struktur hierarki class

# Polimorfisme (Polymorphism) untuk fleksibilitas

# Menyediakan dua mode antarmuka (Console dan GUI) untuk fleksibilitas pengguna

# Mengimplementasikan operasi CRUD lengkap dalam pengelolaan data inventaris

# Memahami mekanisme penyimpanan data persisten menggunakan file teks

# Membuat sistem pelaporan statistik untuk monitoring kondisi dan stok tanaman

# Mengembangkan GUI modern menggunakan Java Swing dengan desain yang intuitif dan menarik

## **1.3 Tujuan Pembuatan Aplikasi**

# Tujuan dari pembuatan aplikasi SPIM (Sistem Pengelolaan Inventaris Tanaman Hias) adalah:

# Merancang dan mengimplementasikan aplikasi pengelolaan inventaris tanaman hias berbasis Java dengan struktur program yang baik.

# Menerapkan konsep Object Oriented Programming secara nyata dalam pembuatan aplikasi, meliputi:

# Enkapsulasi (Encapsulation) untuk melindungi data dan memberikan kontrol akses

# Pewarisan (Inheritance) untuk membuat struktur hierarki class yang terorganisir

# Polimorfisme (Polymorphism) untuk memberikan fleksibilitas dan extensibility

# Mempermudah proses pencatatan dan pengelolaan data inventaris tanaman hias dengan sistem yang terorganisir.

# Menyediakan aplikasi inventaris yang mudah dipahami, digunakan, dan dapat dikembangkan lebih lanjut.

# Menghasilkan output data inventaris yang rinci, terstruktur, dan informatif dalam bentuk tabel dan laporan statistik.

# Memahami mekanisme penyimpanan data persisten menggunakan file teks (TXT) sebagai media penyimpanan sederhana.

# Mengimplementasikan operasi CRUD (Create, Read, Update, Delete) lengkap dalam pengelolaan data inventaris.

# Membuat sistem pelaporan statistik untuk monitoring kondisi dan stok tanaman secara real-time.

# Mengembangkan user interface berbasis console yang user-friendly, intuitif, dan mudah digunakan.

## **1.4 Manfaat Pembuatan Aplikasi**

# Manfaat dari pembuatan aplikasi SPIM ini meliputi:

### 1.4.1 Manfaat Akademik

Aplikasi ini dapat digunakan sebagai sarana pembelajaran untuk:

1. Memahami implementasi enkapsulasi dalam melindungi data
2. Mempelajari konsep pewarisan untuk struktur hierarki class
3. Menguasai penerapan polimorfisme untuk fleksibilitas program
4. Memahami pengembangan dual interface (Console dan GUI)
5. Mempelajari Java Swing untuk pembuatan antarmuka grafis
6. Memahami file I/O operations dan data persistence

### 1.4.2 Manfaat Praktis

Memberikan solusi nyata untuk:

1. Mempermudah pencatatan dan tracking stok tanaman
2. Monitoring kondisi kesehatan tanaman secara sistematis
3. Mengurangi risiko kesalahan pencatatan data
4. Meningkatkan efisiensi dalam pengelolaan inventaris
5. Memudahkan pembuatan laporan dan analisis statistik
6. Memberikan pilihan antarmuka sesuai preferensi pengguna

### 1.4.3 Manfaat Pengembangan Sistem

Aplikasi ini dapat dijadikan dasar untuk:

1. Sistem berbasis database relasional (MySQL, PostgreSQL)
2. Aplikasi web menggunakan framework Java (Spring Boot)
3. Aplikasi mobile Android
4. Integrasi dengan sistem Point of Sale (POS)
5. Penambahan fitur foto tanaman dan barcode scanner
6. Implementasi sistem notifikasi untuk perawatan tanaman.

# **BAB II: DASAR TEORI**

## **2.1 Pengertian Aplikasi**

Aplikasi merupakan perangkat lunak (software) yang dirancang untuk membantu pengguna dalam menyelesaikan suatu pekerjaan tertentu atau mencapai tujuan spesifik. Aplikasi dibuat untuk mempermudah aktivitas manusia dalam berbagai bidang, mulai dari pendidikan, bisnis, hiburan, hingga pengelolaan data.

Dalam konteks ini, **aplikasi SPIM** merupakan perangkat lunak berbasis Java yang berfungsi untuk membantu pengelolaan data inventaris tanaman hias. Aplikasi ini tersedia dalam dua mode: **Console Mode** untuk antarmuka berbasis teks yang efisien, dan **GUI Mode** untuk antarmuka grafis yang lebih visual dan modern.

## **2.2 Konsep Object Oriented Programming (OOP)**

Object Oriented Programming (OOP) adalah paradigma pemrograman yang berfokus pada konsep objek dan kelas (class). OOP bertujuan untuk membuat program lebih terstruktur, modular, mudah dipelihara, dan mudah dikembangkan.

### **2.2.1 Enkapsulasi (Encapsulation)**

**Enkapsulasi** adalah konsep penyembunyian detail implementasi internal suatu objek dan hanya menyediakan akses terbatas melalui method public.

**Cara Penerapan:**

* Membuat atribut class dengan modifier akses private
* Menyediakan getter methods untuk membaca nilai atribut
* Menyediakan setter methods dengan validasi
* Melakukan validasi data sebelum mengubah nilai

**Keuntungan:**

* Data lebih aman dari akses tidak sah
* Kontrol penuh terhadap bagaimana data diakses
* Dapat melakukan validasi sebelum data diubah
* Implementasi internal dapat diubah tanpa mempengaruhi kode luar

### **2.2.2 Pewarisan (Inheritance)**

**Pewarisan** adalah mekanisme dimana sebuah class dapat mewarisi atribut dan method dari class lain.

**Keuntungan:**

* Mengurangi duplikasi kode
* Memudahkan penambahan fitur baru
* Struktur kode lebih hierarkis dan terorganisir
* Mendukung reusability

### **2.2.3 Polimorfisme (Polymorphism)**

**Polimorfisme** adalah kemampuan objek untuk memiliki banyak bentuk. Satu method dapat menampilkan informasi berbeda sesuai tipe objek.

**Keuntungan:**

* Kode lebih fleksibel
* Mudah untuk extensibility
* Mendukung dynamic binding
* Interface yang seragam untuk berbagai implementasi

# **BAB III: Ruang Lingkup Aplikasi**

## **3.1 Ruang Lingkup Aplikasi**

Aplikasi SPIM (Sistem Pengelolaan Inventaris Tanaman Hias) adalah aplikasi berbasis console yang dirancang menggunakan bahasa pemrograman Java dan berjalan pada IntelliJ IDEA atau Java IDE lainnya. Aplikasi ini berfungsi untuk:

### Fitur Utama:

1. Menambah Data Tanaman - Memasukkan data tanaman baru ke dalam inventaris dengan informasi lengkap (ID, nama, jenis, stok, kondisi, dan karakteristik khusus)
2. Melihat Data Tanaman - Menampilkan seluruh data inventaris dalam format tabel yang rapi dan terstruktur
3. Update Data Tanaman - Mengubah atau memperbarui informasi tanaman yang sudah ada dalam sistem
4. Menghapus Data Tanaman - Menghapus data tanaman dari inventaris dengan konfirmasi untuk keamanan
5. Laporan Inventaris - Menampilkan statistik lengkap berupa total stok, breakdown kondisi tanaman, dan analisis inventaris
6. Penyimpanan Otomatis - Setiap perubahan data otomatis tersimpan ke file data\_tanaman.txt

### **Tipe Tanaman yang Didukung:**

1. Tanaman Biasa - Tanaman standar tanpa spesifikasi khusus
2. Tanaman Hias Indoor - Dengan atribut kebutuhan cahaya (Low/Medium/High)
3. Tanaman Hias Outdoor - Dengan atribut ketahanan terhadap hujan (Tahan/Tidak Tahan)

Aplikasi ini bertujuan untuk memberikan solusi praktis dalam pengelolaan inventaris tanaman dengan memanfaatkan konsep pemrograman berorientasi objek yang terstruktur dan efisien.

**3.2 Spesifikasi Aplikasi**

pesifikasi Fungsional

* Menambah data tanaman (Create)
* Menampilkan seluruh data tanaman (Read)
* Memperbarui data tanaman (Update)
* Menghapus data tanaman (Delete)
* Menampilkan laporan statistik inventaris
* Penyimpanan data otomatis ke file .txt

### 

### Spesifikasi Non-Fungsional

* Bahasa pemrograman: Java
* IDE: IntelliJ IDEA
* Antarmuka: Console dan Java Swing (GUI)
* Penyimpanan data: File teks (CSV sederhana)
* Konsep OOP: Enkapsulasi, Pewarisan, Polimorfisme

### Jenis Tanaman

* Tanaman Biasa
* Tanaman Hias Indoor (atribut kebutuhan cahaya)
* Tanaman Hias Outdoor (atribut ketahanan hujan)

**3.3 Rancangan Antarmuka**

Rancangan antarmuka aplikasi SPIM terdiri dari dua jenis:

1. Antarmuka Console (CLI)
   * Menu berbasis teks
   * Navigasi menggunakan input angka
   * Tampilan tabel data inventaris
2. Antarmuka GUI (Java Swing)
   * Form input data tanaman
   * Tabel inventaris dengan tampilan visual
   * Tombol aksi CRUD
   * Laporan inventaris dalam bentuk visual

**3.4 Diagram Proses Aplikasi**

**3.4.1 Flowchart Menu Utama:**

[START]

↓

[Inisialisasi InventoryManager]

↓

[Load Data dari File]

↓

[Tampilkan Menu Utama]

├─→ [1. Tambah Data] → [Input Data] → [Simpan ke List & File]

├─→ [2. Lihat Data] → [Tampilkan Semua Data]

├─→ [3. Update Data] → [Input ID] → [Update Data] → [Simpan]

├─→ [4. Hapus Data] → [Input ID] → [Konfirmasi] → [Hapus & Simpan]

├─→ [5. Laporan] → [Generate Statistik] → [Tampilkan Laporan]

└─→ [6. Keluar] → [Save Data] → [END]

**3.4.2 Flowchart Proses Tambah Data:**

[START Tambah Data]

↓

[Pilih Tipe Tanaman]

├─→ [1. Tanaman Biasa]

├─→ [2. Indoor]

└─→ [3. Outdoor]

↓

[Input Data Umum]

├─ ID Tanaman

├─ Nama

├─ Stok

└─ Kondisi

↓

[Validasi Input]

├─→ [Valid?] ─NO→ [Tampilkan Error] → [Kembali ke Input]

└─→ [YES]

↓

[Input Atribut Khusus]

├─→ [Indoor] → [Kebutuhan Cahaya]

└─→ [Outdoor] → [Tahan Hujan]

↓

[Buat Object Tanaman]

↓

[Tambah ke List]

↓

[Simpan ke File]

↓

[Tampilkan Konfirmasi]

↓

[END]

**3.4.3 Use Case Diagram**

| User/Pengguna |

+---------------+---------------+

| | |

[Tambah Data] [Lihat Data] [Update Data]

| | |

+-------+-------+-------+-------+

| |

[Hapus Data] [Lihat Laporan]

|

+-------+-------+

| |

[Console Mode] [GUI Mode]

+-------+-------+

|

[InventoryManager]

|

+-------+-------+

| |

[File Storage] [Data Validation]

**3.4.4 Class Diagram**

+------------------+

| Tanaman |

+------------------+

| -idTanaman: String

| -namaTanaman: String

| -stok: int

| -kondisi: String

+------------------+

| +getter/setter

| +tampilkanInfo()

| +toString()

+------------------+

▲

|

+---------+---------+

| |

+-------------------+ +--------------------+

| TanamanHiasIndoor | | TanamanHiasOutdoor |

+-------------------+ +--------------------+

| -kebutuhanCahaya | | -tahanHujan: boolean

+-------------------+ +--------------------+

| +tampilkanInfo() | | +tampilkanInfo()

+-------------------+ +--------------------+

+----------------------+

| InventoryManager |

+----------------------+

| -daftarTanaman: List<Tanaman>

| -FILE\_NAME: String

+----------------------+

| +tambahTanaman()

| +lihatSemuaTanaman()

| +updateTanaman()

| +hapusTanaman()

| +laporanInventaris()

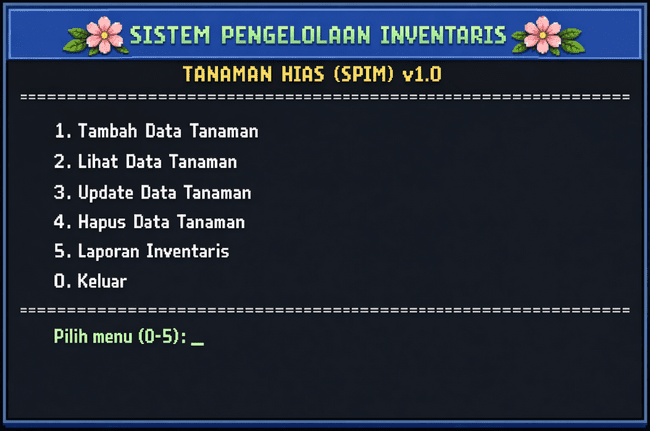
| +simpanKeFile()

| +muatDariFile()

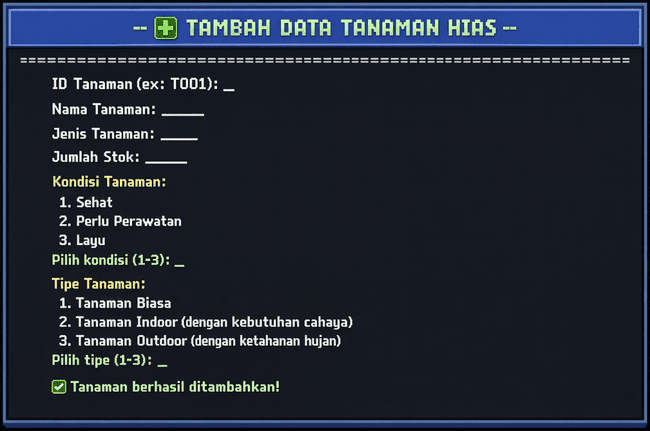
+----------------------+

## **3.5 Desain / Rancangan Aplikasi Dalam Console & GUI**

### **Sketsa Visual Aplikasi**



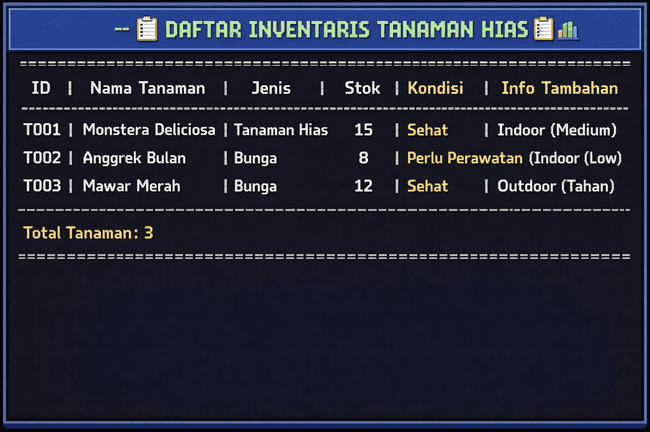
### **Sketsa Form Input Data Tanaman**



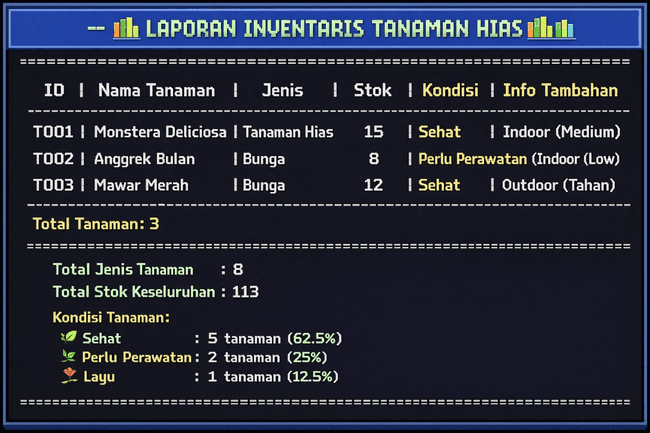
### 

### 

### **Sketsa Tampilan Data Inventaris**



### **Sketsa Laporan Inventaris**



## **3.5 Konsep Object-Oriented Programming (OOP)**

## **3.5.1 Enkapsulasi (Encapsulation)**

Enkapsulasi adalah konsep menyembunyikan detail implementasi dan hanya menyediakan akses terbatas melalui method public. Dalam aplikasi SPIM:

* Semua atribut class dibuat private
* Akses data melalui getter dan setter methods
* Validasi data dilakukan dalam setter

Keuntungan:

* Data lebih aman dari akses tidak sah
* Kontrol penuh terhadap modifikasi data
* Memudahkan maintenance kode

### **3.5.2 Pewarisan (Inheritance)**

Pewarisan adalah mekanisme dimana sebuah class dapat mewarisi atribut dan method dari class lain. Dalam aplikasi SPIM:

* Class Tanaman sebagai parent class (superclass)
* Class TanamanHiasIndoor sebagai child class
* Class TanamanHiasOutdoor sebagai child class

Keuntungan:

* Mengurangi duplikasi kode
* Memudahkan penambahan fitur baru
* Struktur kode lebih hierarkis dan terorganisir

### **3.5.3 Polimorfisme (Polymorphism)**

Polimorfisme adalah kemampuan objek untuk memiliki banyak bentuk. Dalam aplikasi SPIM:

* Method tampilkanInfo() di-override di setiap child class
* Satu method dapat menampilkan informasi berbeda sesuai tipe objek
* Runtime menentukan method mana yang dijalankan

Keuntungan:

* Kode lebih fleksibel
* Mudah untuk ekstensibilitas
* Mendukung dynamic binding

**BAB IV: IMPLEMENTASI**

**4.1 Alat dan Bahan**

### 1. Perangkat Lunak:

* Java Development Kit (JDK) - Version 8 atau lebih tinggi
* IntelliJ IDEA - IDE untuk development Java
* Text Editor - Untuk membuka file data\_tanaman.txt
* Git (Opsional) - Untuk version control

### 2. Perangkat Keras:

* Laptop/PC dengan spesifikasi minimal:
  + Processor: Intel Core i3 atau setara
  + RAM: 4GB (Minimum), 8GB (Recommended)
  + Storage: 500MB free space
  + javax.swing. - GUI components

### 3. Library yang Digunakan:

* java.util.Scanner - Untuk input console
* java.util.List & ArrayList - Untuk koleksi data
* java.io. - Untuk file I/O operations

## **4.2 Langkah-Langkah Pembuatan**

### 1. Desain Program:

* Menentukan struktur class utama (Tanaman, TanamanHiasIndoor, TanamanHiasOutdoor)
* Membuat class InventoryManager untuk pengelolaan data
* Merancang alur program dan user interface

### 2. Implementasi Class:

Tahap 1: Membuat parent class Tanaman dengan enkapsulasi

* Definisikan atribut private
* Buat constructor, getter, dan setter
* Implementasikan method tampilkanInfo() dan toString()

Tahap 2: Membuat child classes dengan pewarisan

* TanamanHiasIndoor extends Tanaman
* TanamanHiasOutdoor extends Tanaman
* Override method yang diperlukan

Tahap 3 : Membuat Tanaman GUI

konsep GUI yang diterapkan :

* MVC Pattern: Pemisahan Model (InventoryManager), View (GUI), Controller (Event Handlers)
* Event-Driven Programming: Button actions dan user interactions
* Custom Rendering: Color coding untuk kondisi tanaman
* CardLayout: Navigasi antar panel
* Data Binding: Sinkronisasi antara model dan view

Tahap 4: Membuat InventoryManager untuk polimorfisme

* Implementasi List<Tanaman> untuk menyimpan berbagai tipe
* Buat method CRUD lengkap
* Implementasi file I/O untuk persistent storage

### 3. Implementasi Main Program:

* Membuat menu utama interaktif
* Implementasi input validation
* Menambahkan error handling

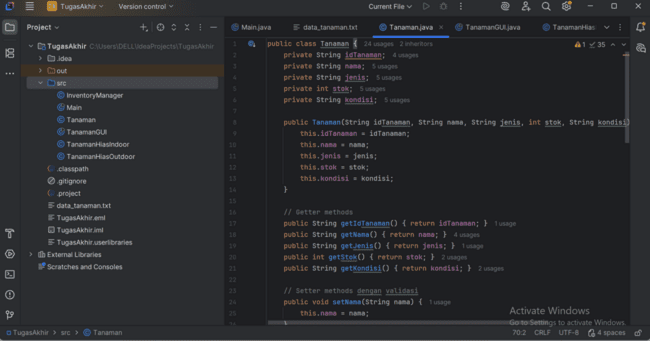
### 4. Testing dan Debugging:

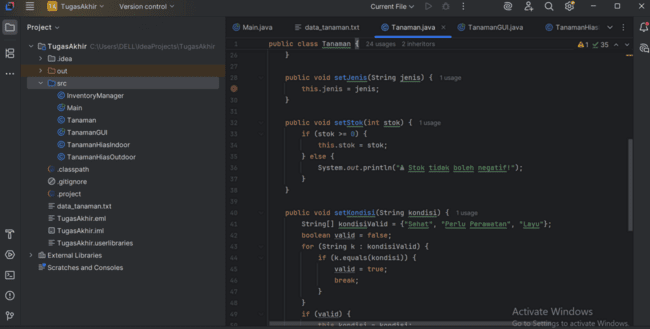
* Uji setiap fitur secara individual
* Test dengan berbagai skenario input
* Perbaiki bug yang ditemukan

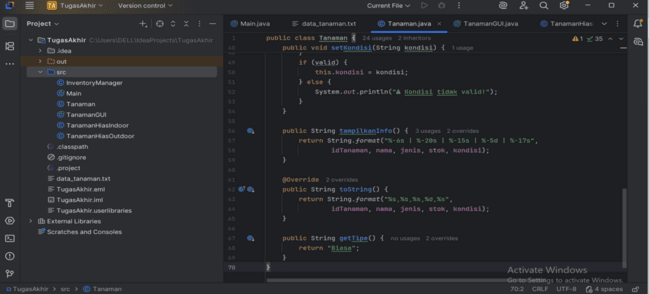
## **4.3 Kode Program Console & GUI**

Berikut adalah progress pengkodean yang telah dilakukan hingga selesai:

### **4.3.1 Class Tanaman.java (Parent Class - Enkapsulasi)**



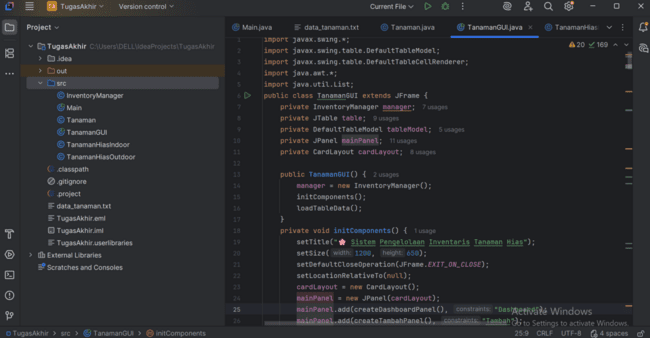


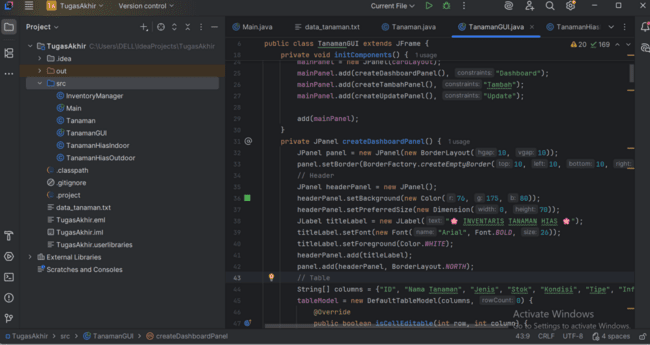


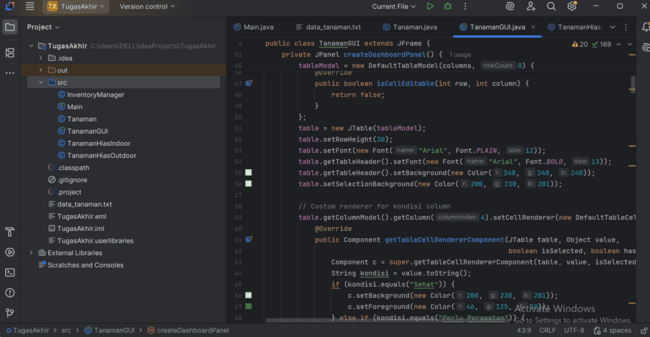
**Penjelasan:**

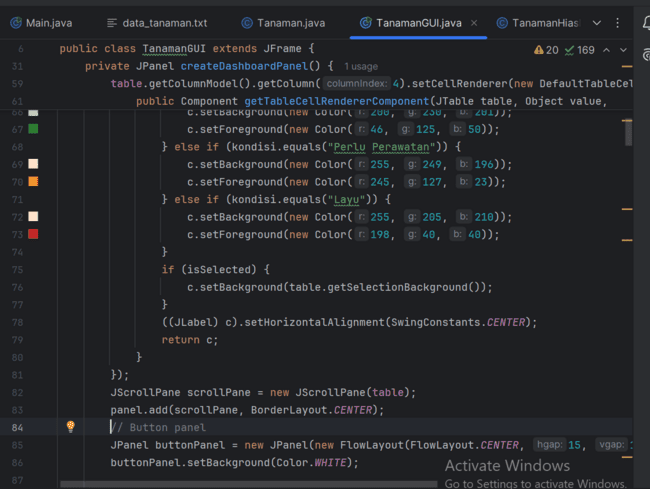
* Semua atribut menggunakan modifier private untuk enkapsulasi
* Akses data hanya melalui getter/setter methods
* Validasi data pada setter (contoh: stok tidak boleh negatif)
* Method tampilkanInfo() untuk display terformat
* Method toString() untuk penyimpanan file

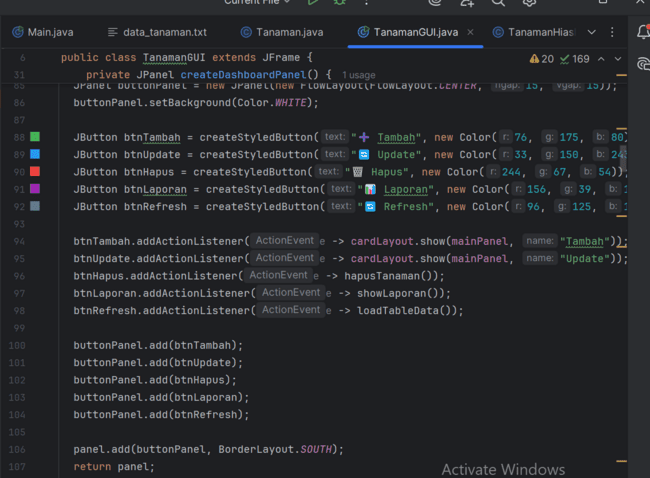
### **4.3.2Class Tanaman GUI.java**





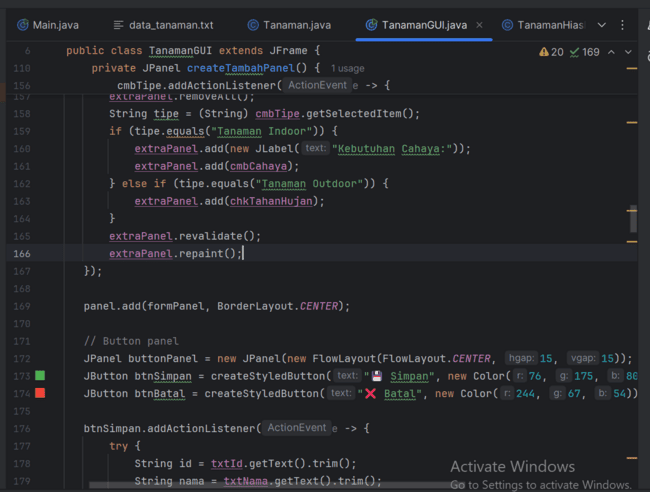


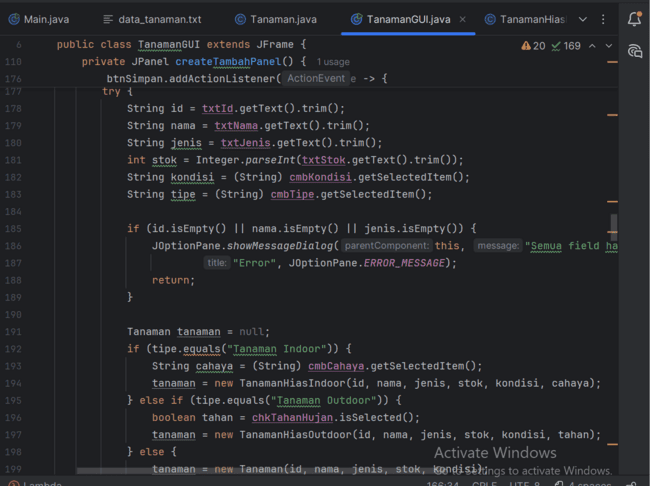


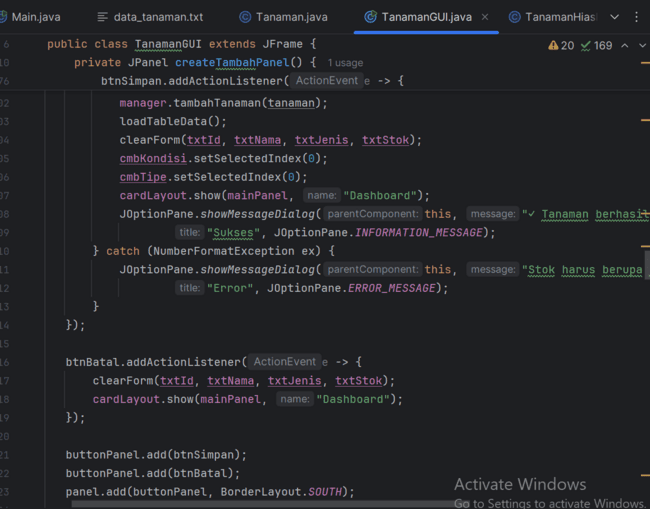


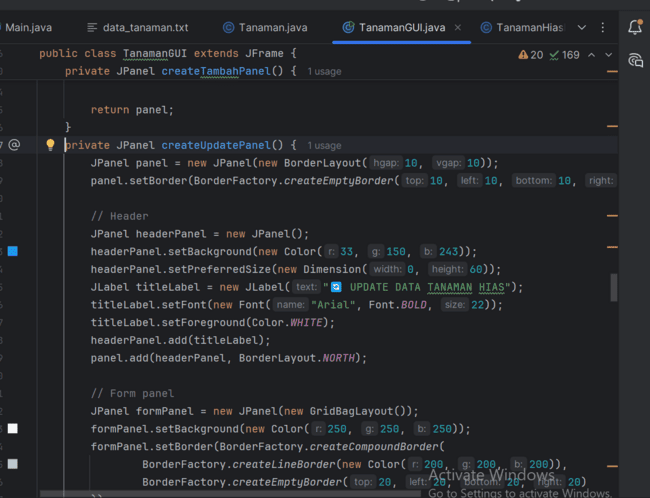
### 

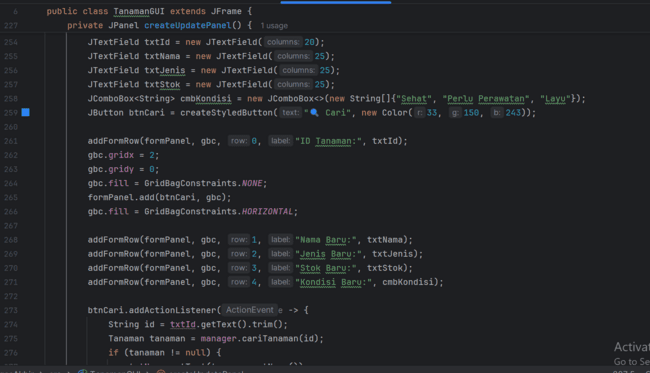
### 

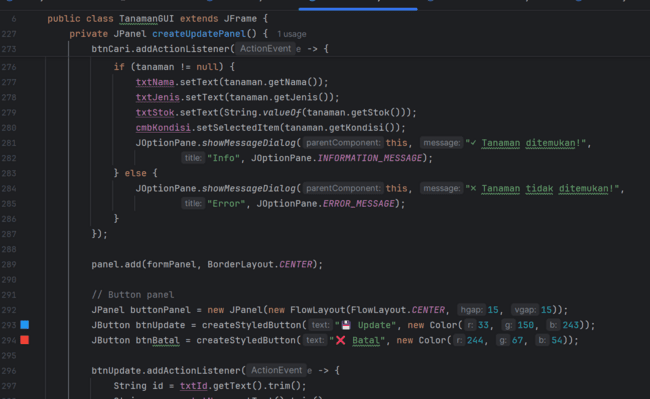


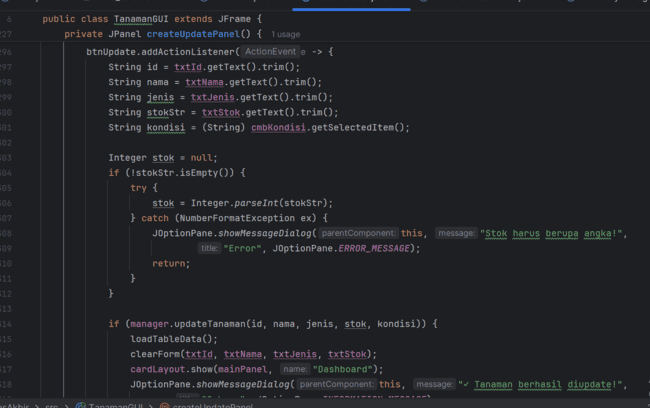


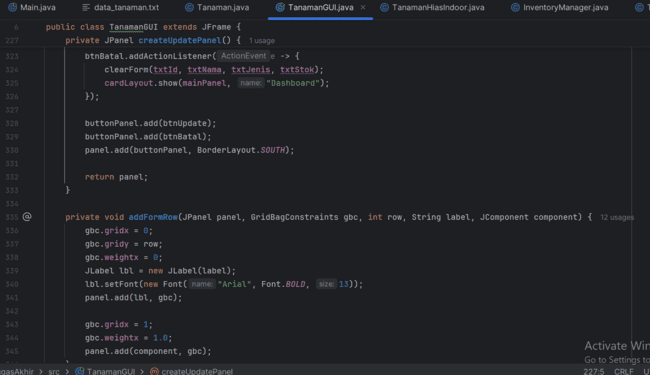


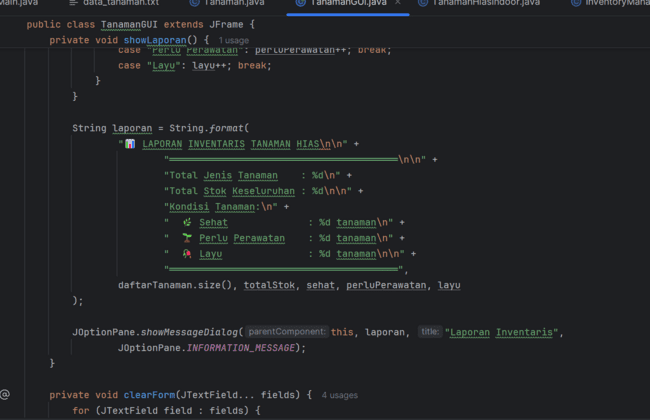








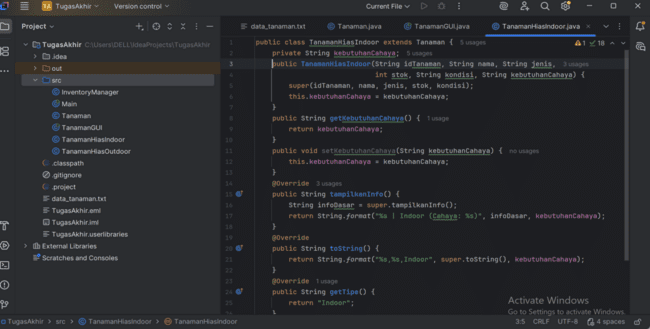




**Penjelasan :**

* Mendemonstrasikan Inheritance melalui class Tanaman & turunannya
* Menggunakan Polymorphism saat menyimpan dan memproses objek
* Menampilkan perilaku berbeda di GUI tanpa mengubah struktur utama
* Memisahkan tampilan (GUI) dan logika data dengan baik

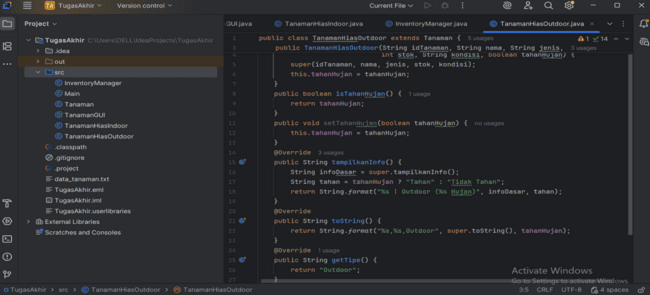
### **4.3.3 Class TanamanHiasIndoor.java (Child Class - Pewarisan)**

****

**Penjelasan:**

* extends Tanaman - Mewarisi semua atribut dan method parent
* super() - Memanggil constructor parent class
* Menambahkan atribut khusus: kebutuhanCahaya
* @Override - Override method untuk behavior khusus
* Mendemonstrasikan konsep inheritance dengan jelas

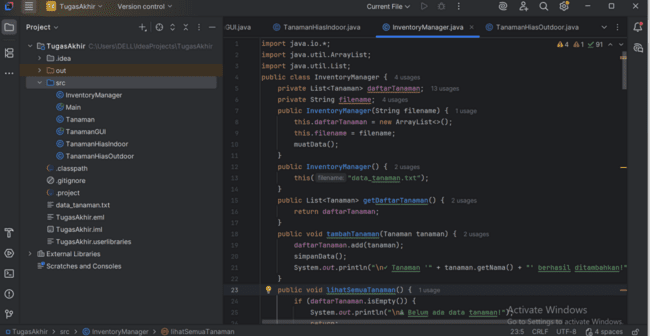
### **4.3.4 Class TanamanHiasOutdoor.java (Child Class - Pewarisan)**

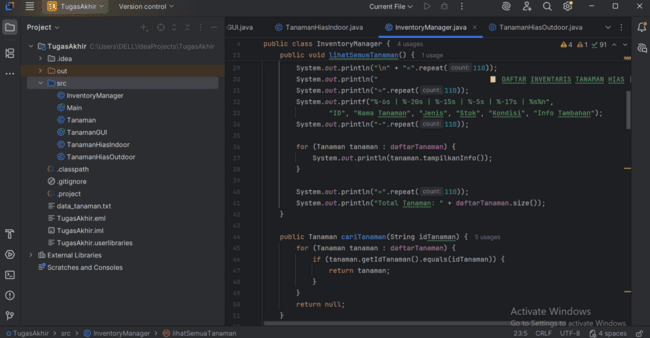


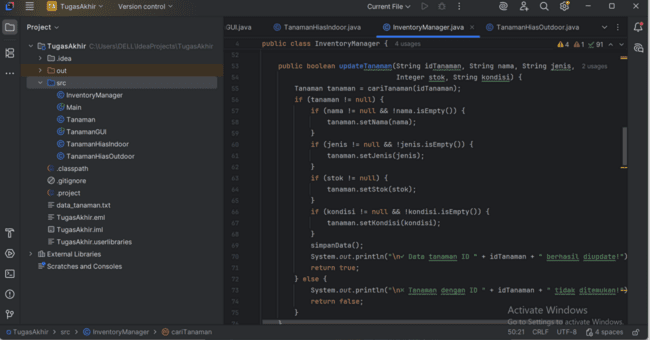
**Penjelasan:**

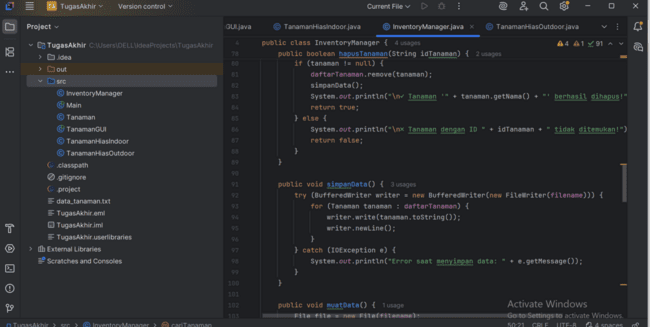
* extends Tanaman - Class kedua yang mewarisi parent
* Atribut tambahan: tahanHujan (boolean)
* Override method dengan implementasi berbeda
* Menunjukkan polymorphism - method sama, behavior berbeda

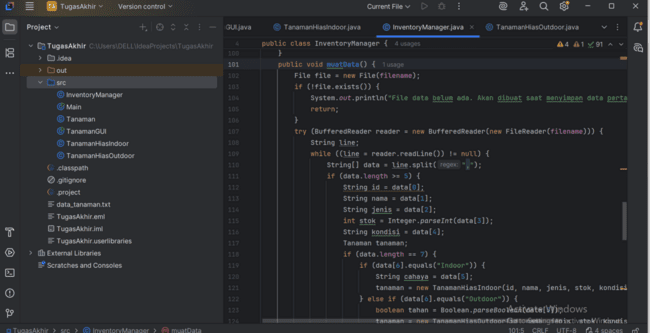
### **4.3.5 Class InventoryManager.java (Manager Class - Polimorfisme)**

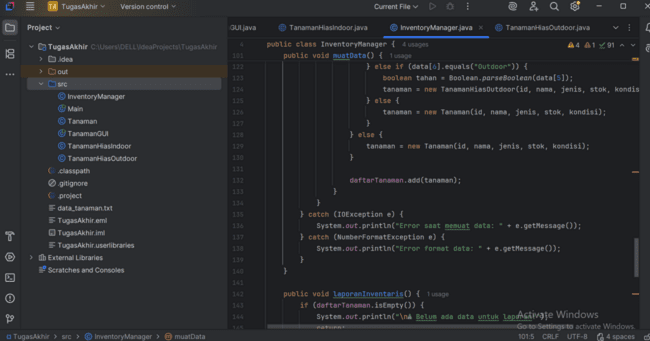


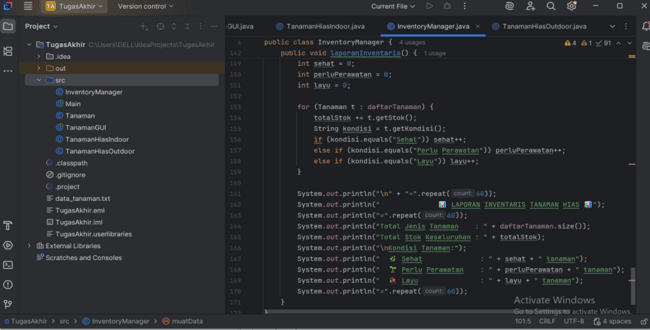












Penjelasan:

* List<Tanaman> - Polimorfisme dalam koleksi data
* Method tambahTanaman() menerima berbagai tipe objek
* Loop polimorfis pada lihatSemuaTanaman()
* Implementasi lengkap CRUD operations
* File I/O untuk persistent storage
* Error handling dengan try-catch

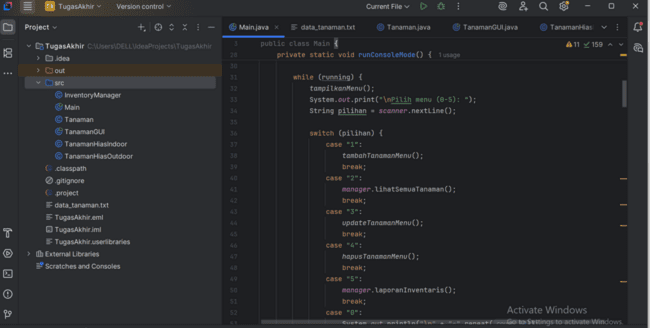
### 

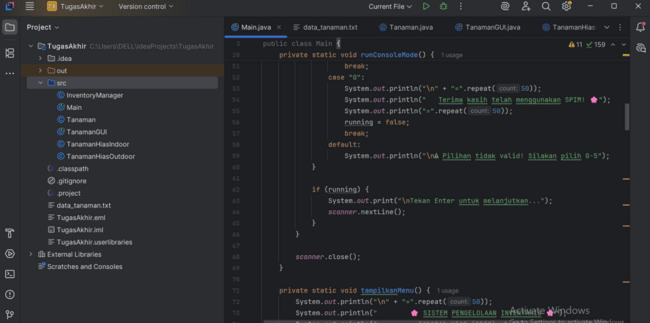
### 

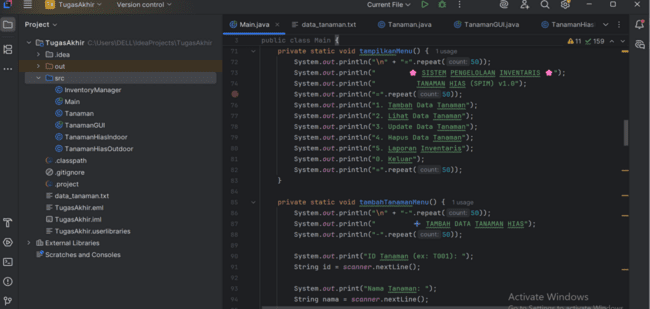
### 

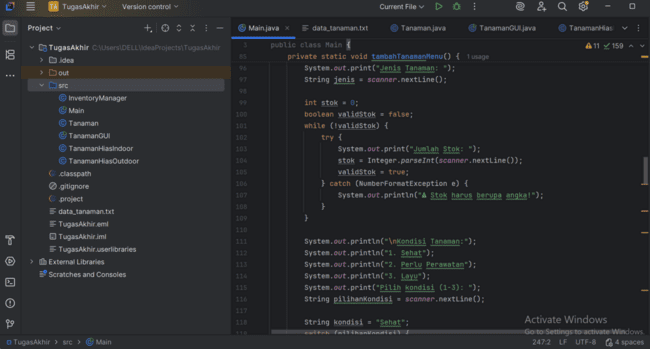
### **4.3.6 Class Main.java (Program Utama)**

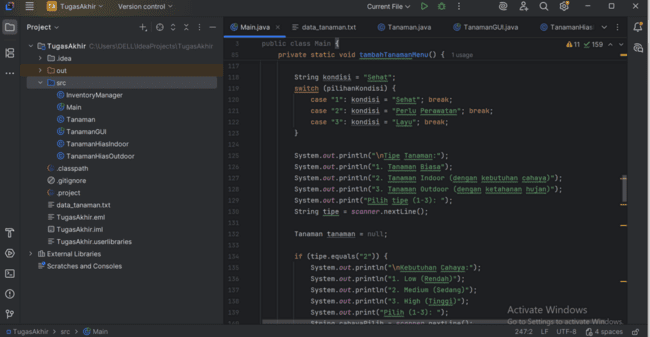
****

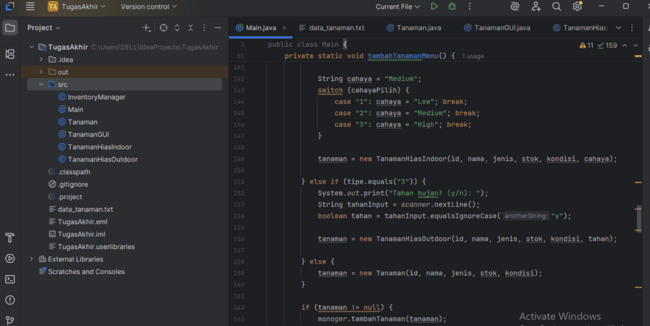
****

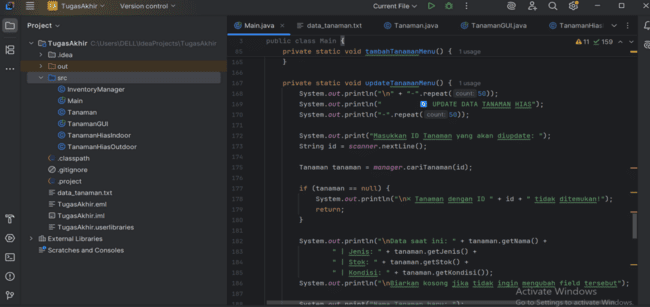
****

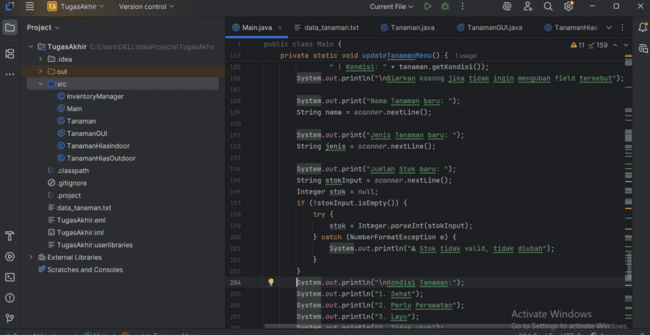
****

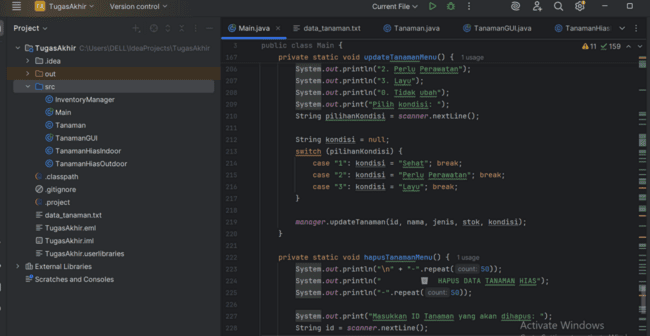
****

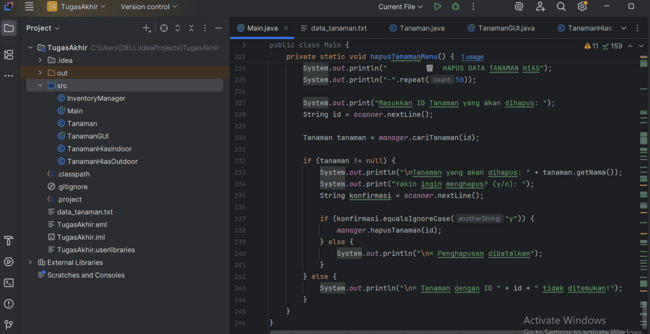
****

****

****

****

****

****

**Penjelasan:**

* Format CSV sederhana
* Mendukung 3 tipe tanaman
* Data contoh untuk testing
* penambahan data GUI

### 

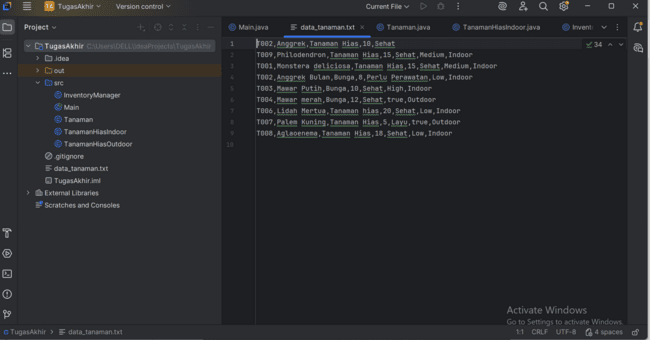
### 

### 

### 

### 

### **4.3.7 Data txt**



**Penjelasan :**

Untuk menyimpan semua data.

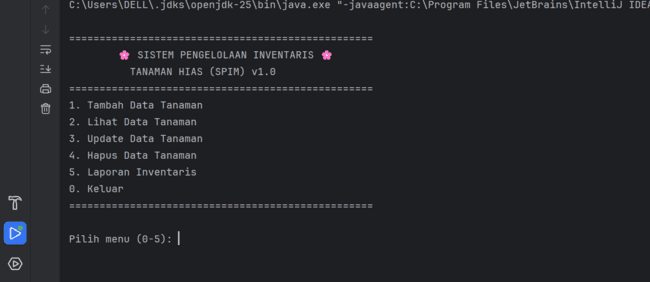
**BAB V: HASIL DAN PEMBAHASAN**

## **5.1 Hasil Implementasi Dari Console dan GUI**

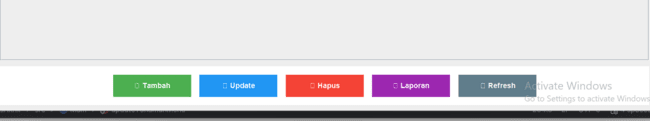
Aplikasi SPIM (Sistem Pengelolaan Inventaris Tanaman Hias) berhasil diimplementasikan dengan sempurna menggunakan IntelliJ IDEA. Semua fitur yang dirancang telah berfungsi dengan baik.

### **5.1.1 Screenshot Menu Utama**

**Console**



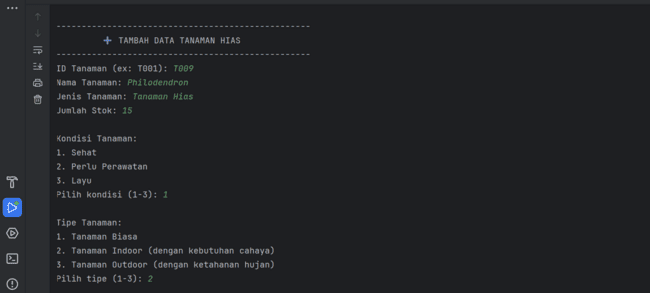
**GUI**

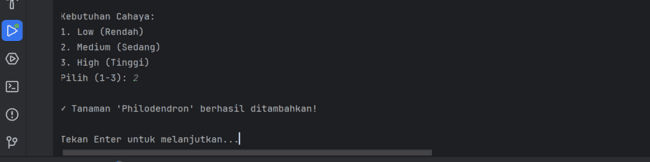


### 

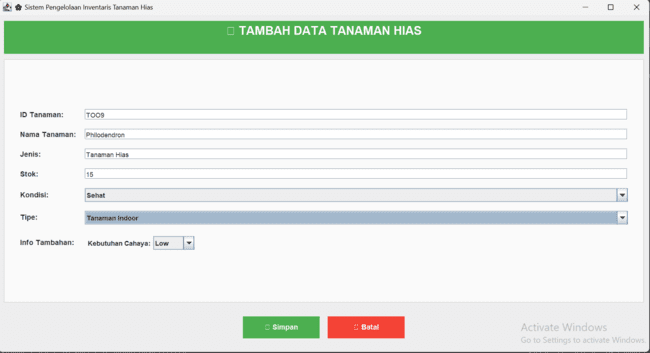
### **5.1.2 Screenshot Tambah Data Tanaman**

**Console**



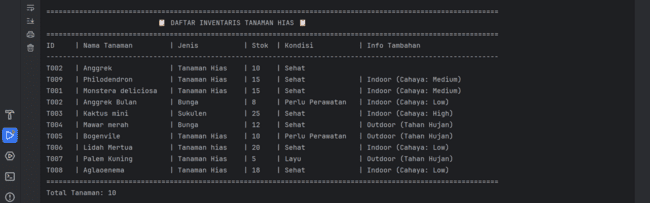


**GUI**

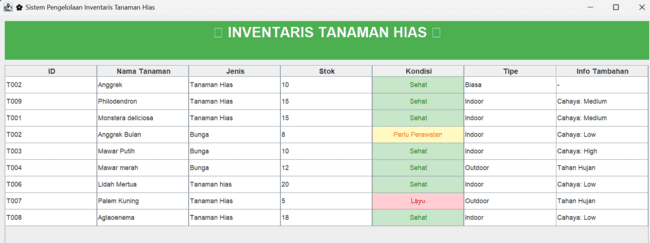
****

### **5.1.3 Screenshot Lihat Data Tanaman**

**Console**

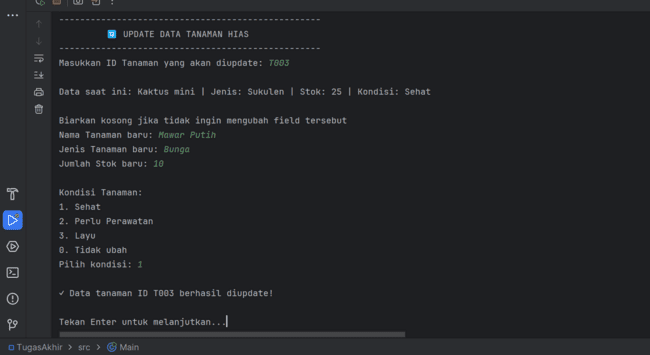


**GUI**

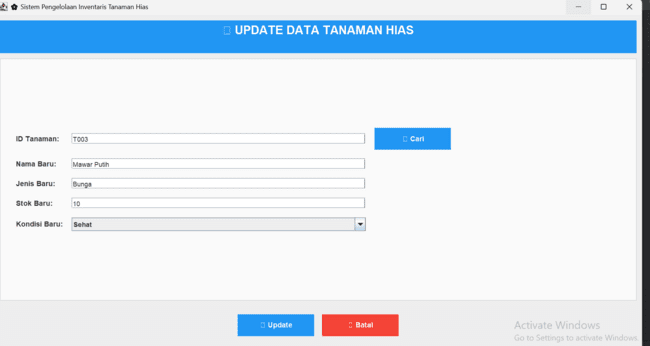
****

### **5.1.4 Screenshot Update Tanaman**

**Console**

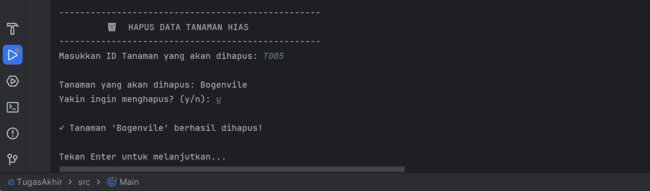


**GUI**

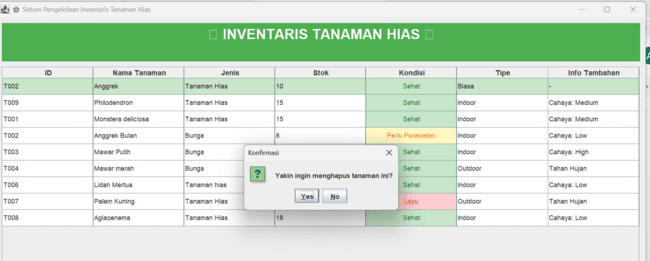
****

### **5.1.5 Screenshot Hapus Data Tanaman**

**Console**



**GUI**

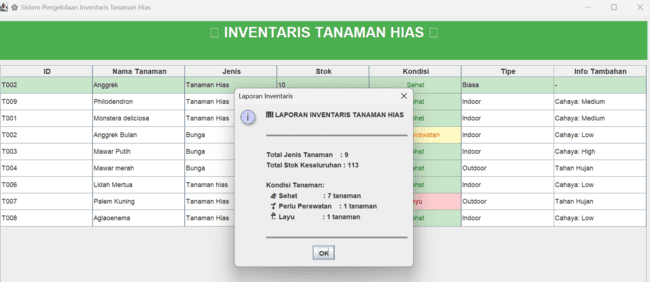


### **5.1.6 Screenshot Laporan Inventaris**

**Console**

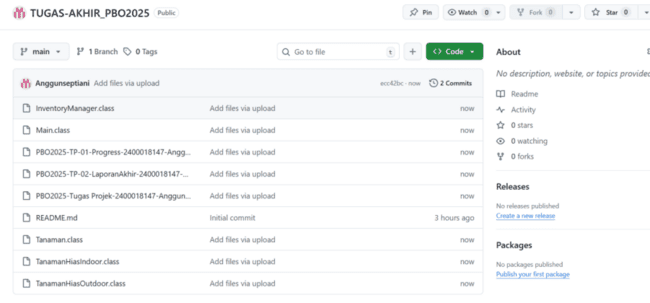


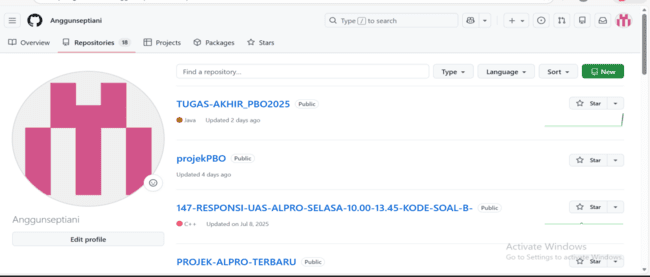
**GUI**

****

**5.2 Screenshot Status Unggah di GitHub**

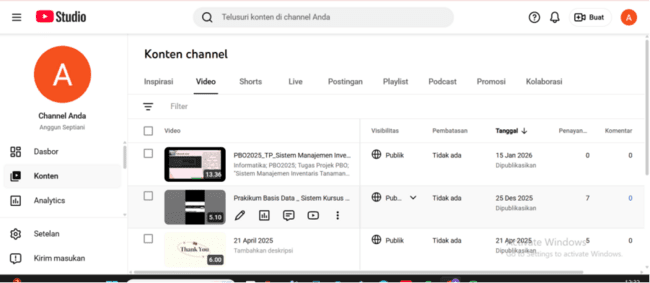
**Link Project Github :** <https://github.com/Anggunseptiani/TUGAS-AKHIR_PBO2025>

****

****

**5.2 Screenshot Status Unggah di Youtube**

**Link Youtube : <https://youtu.be/_c8Wn3oW-II>**

****

## **5.3 Analisis Implementasi**

### **5.3.1 Analisis Konsep OOP**

1. Enkapsulasi:

* Implementasi Sempurna: Semua atribut private dengan getter/setter
* Validasi Data: Setter melakukan pengecekan sebelum update
* Information Hiding: Detail internal tersembunyi dari luar
* Keamanan Data: Data tidak bisa diakses/diubah sembarangan

2. Pewarisan:

* Hierarki Class: Parent (Tanaman) → Children (Indoor, Outdoor)
* Code Reusability: Child class mewarisi semua method parent
* Extensibility: Mudah menambahkan tipe tanaman baru
* Struktur Terorganisir: Logika umum di parent, spesifik di child

3. Polimorfisme:

* Runtime Polymorphism: Method overriding dengan @Override
* Dynamic Binding: Method yang dipanggil ditentukan saat runtime
* Interface Uniform: Satu interface untuk berbagai implementasi
* Flexibility: Mudah menambah tipe baru tanpa ubah existing code

### 

### 

### **5.3.2 Analisis Fungsionalitas**

1. Operasi CRUD:

* Create: Tambah data berfungsi dengan baik
* Read: Tampilan data rapi dan terstruktur
* Update: Modifikasi data berhasil
* Delete: Penghapusan dengan konfirmasi

2. Persistent Storage:

* Data tersimpan otomatis ke file
* Data ter-load saat aplikasi dimulai
* Format CSV mudah dibaca
* Data Integrity: Tidak ada data loss saat aplikasi ditutup

3. Dual Interface:

* Console Mode: Efisien, lightweight, cocok untuk server/remote
* GUI Mode: Visual, user-friendly, cocok untuk desktop
* Shared Backend: Kedua mode menggunakan InventoryManager yang sama
* Data Consistency: Perubahan dari satu mode terlihat di mode lain

## **5.4 Kelebihan Aplikasi**

1. Struktur Kode yang Baik:
   * Implementasi OOP yang benar dan jelas
   * Setiap class memiliki tanggung jawab spesifik
   * Mudah untuk maintenance dan development
2. Dual Interface yang Fleksibel:
   * Console mode untuk efisiensi dan remote access
   * GUI mode untuk user experience yang lebih baik
   * Shared backend untuk konsistensi data
   * Easy mode switching melalui launcher
3. Fleksibel dan Extensible:
   * Mudah menambahkan tipe tanaman baru
   * Struktur yang mendukung pengembangan fitur
   * Polimorfisme memudahkan ekstensibilitas

## **5.5 Kelemahan Aplikasi**

1. Belum Menggunakan Database:
   * Penyimpanan masih berbasis file teks
   * Tidak optimal untuk data besar
   * Kurang aman untuk data sensitif
2. Fitur pencarian :
   * earch hanya by ID
   * Belum ada filter berdasarkan kondisi/tipe
   * sorting tabel masih manual
3. Validasi Input Sederhana:
   * Pengecekan duplikasi ID belum optimal
   * Format ID belum terstandarisasi
   * Belum ada validasi format data kompleks

**5.6 Analisis Pengerjaan Proyek**

### **Analisis Waktu**

Pengerjaan proyek dilakukan secara bertahap mulai dari perancangan, pengkodean, pengujian, hingga dokumentasi. Waktu pengerjaan dapat dikelola dengan baik sesuai dengan jadwal Saya selama perkuliahan dan selama minggu tenang..

### **Ketercapaian Spesifikasi**

Seluruh spesifikasi yang direncanakan berhasil diimplementasikan, termasuk dual interface, CRUD, penyimpanan data, dan laporan inventaris.

### **Biaya yang Dibutuhkan**

Tidak ada biaya yang dibutuhkan dalam pengerjaan proyek ini karena seluruh proses pengembangan menggunakan perangkat pribadi dan perangkat lunak gratis (open-source), seperti Java Development Kit (JDK), IntelliJ IDEA Community Edition, serta platform GitHub dan YouTube yang dapat digunakan tanpa biaya.

### Kendala

* Pengelolaan sinkronisasi data antara Console dan GUI
* Debugging error pada file I/O
* Penyesuaian desain GUI agar user-friendly

### Tantangan Masa Depan

* Migrasi penyimpanan ke database
* Penambahan fitur pencarian dan filter
* Pengembangan ke aplikasi web atau mobile

# 

# **BAB VI: PENUTUP**

## **6.1 Kesimpulan**

Berdasarkan implementasi dan analisis yang telah dilakukan, dapat disimpulkan bahwa:

1. **Aplikasi SPIM berhasil diimplementasikan dengan sempurna** dalam dua mode (Console dan GUI), menerapkan tiga konsep utama Object-Oriented Programming:
   * Enkapsulasi untuk melindungi data dengan akses terkontrol
   * Pewarisan untuk membuat hierarki class yang terstruktur
   * Polimorfisme untuk memberikan fleksibilitas dan extensibility
2. **Dual interface berhasil diimplementasikan** dengan:
   * Console mode untuk efisiensi dan remote access
   * GUI mode untuk user experience yang modern
   * Shared backend untuk konsistensi data
   * Easy mode switching melalui launcher
3. **Semua fitur yang dirancang telah berhasil diimplementasikan** dengan baik:
   * Operasi CRUD lengkap (Create, Read, Update, Delete)
   * Persistent storage menggunakan file I/O
   * Sistem pelaporan statistik real-time
   * Input validation dan error handling
   * User-friendly interface (Console & GUI)
4. **Konsep OOP diterapkan secara konsisten** di seluruh aplikasi:
   * Struktur class yang hierarkis dan terorganisir
   * Code reusability melalui inheritance
   * Dynamic binding melalui polymorphism
   * Data protection melalui encapsulation
5. **Aplikasi memiliki nilai praktis** yang dapat digunakan untuk:
   * Media pembelajaran OOP yang konkret
   * Solusi nyata untuk pengelolaan inventaris tanaman
   * Foundation untuk pengembangan sistem yang lebih kompleks
6. **GUI Implementation dengan Java Swing** menunjukkan:
   * Integrasi sukses antara backend dan frontend
   * Modern UI dengan color coding dan styling
   * Event-driven programming yang baik
   * Responsive dan user-friendly design

**Lampiran PPT**

<https://www.canva.com/design/DAG95zXS2QM/XlS6g5Qpn3STxsy9enTnYg/view?utm_content=DAG95zXS2QM&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h285de19c5b>

**Lampiran link Youtube**

youtube **: <https://youtu.be/_c8Wn3oW-II>**