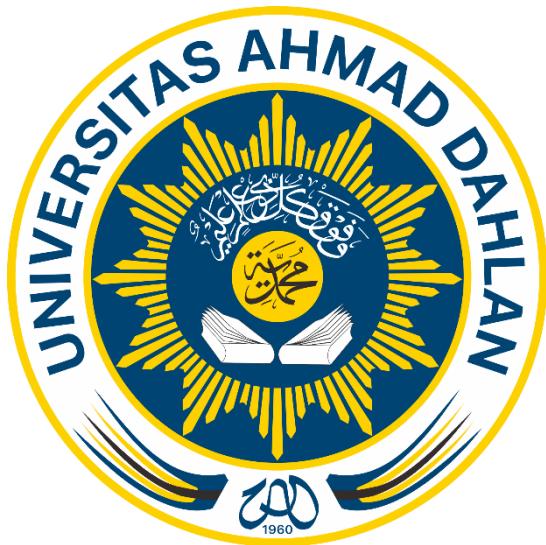


# LAPORAN PROYEK PEMROGRAMAN BERORIENTASI OBJEK

Aplikasi Donasi Barang Layak Pakai Berbasis OOP



Disusun oleh:

Muhammad Yusuf Mubarok (2400018025)

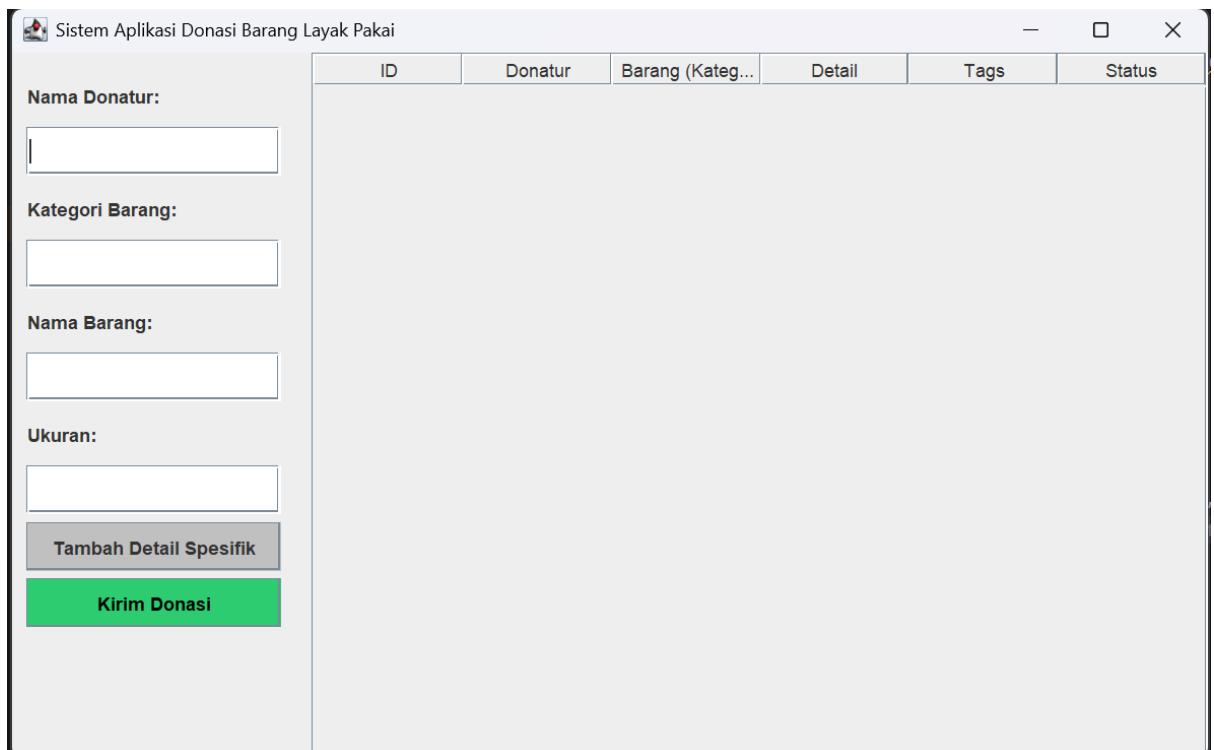
Ghaffar Taturi H. (2400018021)

Adam Januar Fauzi (2400018002)

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS AHMAD DAHLAN  
DAERAH ISTIMEWA YOGYAKARTA**

## A. Identitas Proyek

- **Mata Kuliah : Pemrograman Berorientasi Objek (PBO)**
- **Judul Proyek : Aplikasi Donasi Barang Layak Pakai Berbasis OOP**
- **Bahasa Pemrograman : Java**
- **Platform : Desktop (Console / GUI)**
- **Link Repository GitHub/GitLab :**  
<https://github.com/mubarok010706/project-PBO>
- **Deskripsi singkat : Platform digital untuk menyalurkan barang layak pakai secara sistematis. Dengan menerapkan konsep Encapsulation dan Inheritance, aplikasi ini mempermudah pengelompokan jenis barang donasi serta pelacakan status pengiriman dari donatur hingga ke tangan yang membutuhkan.**



The screenshot shows a window titled "Sistem Aplikasi Donasi Barang Layak Pakai". On the left, there is a form with fields for "Nama Donatur" (Name of Donor), "Kategori Barang" (Item Category), "Nama Barang" (Item Name), and "Ukuran" (Size). Below these fields are two buttons: "Tambah Detail Spesifik" (Add Specific Details) and "Kirim Donasi" (Send Donation). On the right, there is a table header with columns: ID, Donatur, Barang (Kateg...), Detail, Tags, and Status.

ID	Donatur	Barang (Kateg...)	Detail	Tags	Status
----	---------	-------------------	--------	------	--------

---

## B. Persoalan Bisnis dan Deskripsi Proyek

### Persoalan Bisnis

**Banyak masyarakat memiliki barang yang masih layak pakai namun tidak terpakai, sementara di sisi lain terdapat pihak yang membutuhkan barang tersebut. Proses donasi sering kali tidak terdata dengan baik, kurang transparan, dan sulit dikelola secara manual.**

### **Deskripsi Proyek**

**Aplikasi Donasi Barang Layak Pakai Berbasis OOP dikembangkan untuk memfasilitasi proses donasi barang secara terstruktur, mulai dari pendataan donatur, barang yang didonasikan, hingga penerima donasi. Aplikasi ini menerapkan konsep Object-Oriented Programming untuk menghasilkan sistem yang terstruktur, mudah dikembangkan, dan mudah dipelihara.**

---

### **C. Daftar Seluruh Spesifikasi Aplikasi**

- 1. Sistem login pengguna (Donatur dan Penerima)**
  - 2. Pendataan donatur dan penerima**
  - 3. Pendataan barang donasi**
  - 4. Proses donasi barang**
  - 5. Penentuan status donasi**
  - 6. Penyaluran barang ke penerima**
  - 7. Menampilkan informasi donasi**
  - 8. Output aplikasi bersifat dinamis**
  - 9. Penerapan konsep OOP (Inheritance, Encapsulation, Polymorphism)**
- 

### **D. Rancangan Model Diagram UML**

#### **Class Diagram (Deskripsi)**

- User sebagai superclass**
- Donatur dan Penerima sebagai subclass dari User**
- Barang sebagai objek yang didonasikan**
- Donasi sebagai penghubung antara Donatur, Barang, dan Penerima**

#### **Relasi UML:**

- Inheritance: User → Donatur, Penerima**

- **Association:** Donatur ↔ Barang
- **Aggregation:** Donasi ◇– Barang
- **Composition:** Donasi ◆– Donatur
- **Polymorphism:** Method overriding dan overloading

(Lampirkan gambar diagram UML)

---

## E. Rancangan Antar Muka Berbasis GUI

### Deskripsi Rancangan GUI

Rancangan antarmuka aplikasi dibuat dalam bentuk GUI sederhana berbasis desktop. Antarmuka ini dirancang untuk memudahkan pengguna dalam melakukan input data dan melihat hasil donasi barang layak pakai. Walaupun implementasi aplikasi masih berbasis console, rancangan GUI ini disiapkan sebagai acuan pengembangan selanjutnya.

#### Komponen Antarmuka

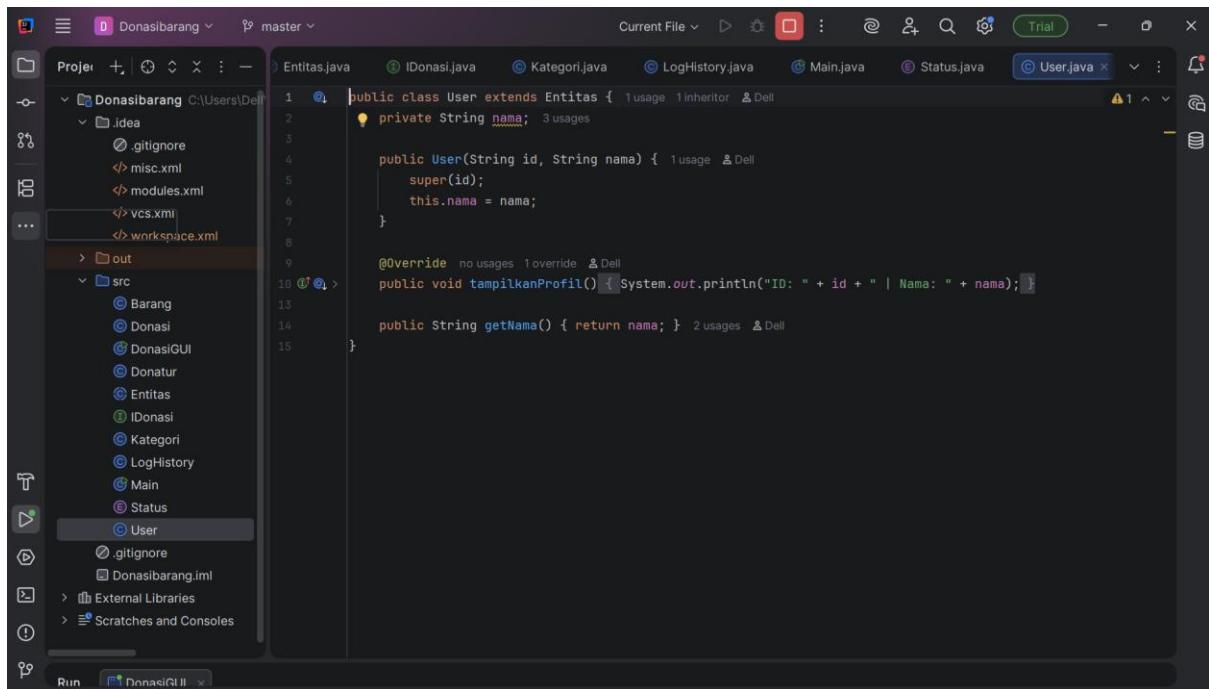
1. **Form Input Donatur**
  - **TextField Nama Donatur**
  - **Tombol Lanjut**
2. **Form Input Penerima**
  - **TextField Nama Penerima**
  - **Tombol Lanjut**
3. **Form Input Barang Donasi**
  - **TextField Nama Barang**
  - **ComboBox Kondisi Barang (Layak Pakai / Sangat Layak)**
  - **Tombol Proses Donasi**
4. **Tampilan Informasi Donasi**
  - **Label Nama Donatur**
  - **Label Nama Penerima**
  - **Label Nama Barang**
  - **Label Status Donasi**

---

## F. Skrip Program dan Penjelasannya

**Aplikasi Donasi barang layak pakai dikembangkan menggunakan konsep OOP dengan beberapa kelas utama, yaitu :**

### 1. Class User



The screenshot shows the IntelliJ IDEA interface with the code editor open to the User.java file. The code defines a class User that extends Entitas. It has a private attribute nama and a constructor that takes id and nama. It also has a method tampilanProfil() which prints the ID and name to the console, and a getName() method that returns the name.

```
public class User extends Entitas { 1 usage 1 inheritor & Dell
    private String nama; 3 usages

    public User(String id, String nama) { 1 usage & Dell
        super(id);
        this.nama = nama;
    }

    @Override no usages 1 override & Dell
    public void tampilanProfil() { System.out.println("ID: " + id + " | Nama: " + nama); }

    public String getName() { return nama; } 2 usages & Dell
}
```

**Berfungsi sebagai superclass yang menyimpan atribut umum pengguna dan method login.**

### 2. Class Donatur

The screenshot shows the IntelliJ IDEA interface with the project 'Donasibarang' open. The left sidebar displays the project structure with packages like .idea, .gitignore, misc.xml, modules.xml, vcs.xml, workspace.xml, out, and src containing classes such as Barang, Donasi, DonasiGUI, Donatur, Entitas, IDonasi, Kategori, LogHistory, Main, Status, and User. The current file is 'Donatur.java', which defines a class 'Donatur' extending 'User'. The code includes methods for displaying the donor's profile and handing over donations.

```
1 public class Donatur extends User { no usages & Dell
2     public Donatur(String id, String nama) { super(id, nama); }
3
4     @Override no usages & Dell
5     public void tampilanProfil() { System.out.println("[Profil Donatur] " + getNama()); }
6
7     // Asosiasi: Berinteraksi dengan objek Donasi
8     public void serahkanDonasi(Donasi<> donasi) { no usages & Dell
9         System.out.println("Donatur " + getNama() + " menyerahkan Donasi ID: " + donasi.getDonasiId());
10    }
11
12    @
13    {
14        System.out.println("Donatur " + getNama() + " menyerahkan Donasi ID: " + donasi.getDonasiId());
15    }
16}
```

Merupakan turunan dari User yang memiliki fungsi untuk mendonasikan barang.

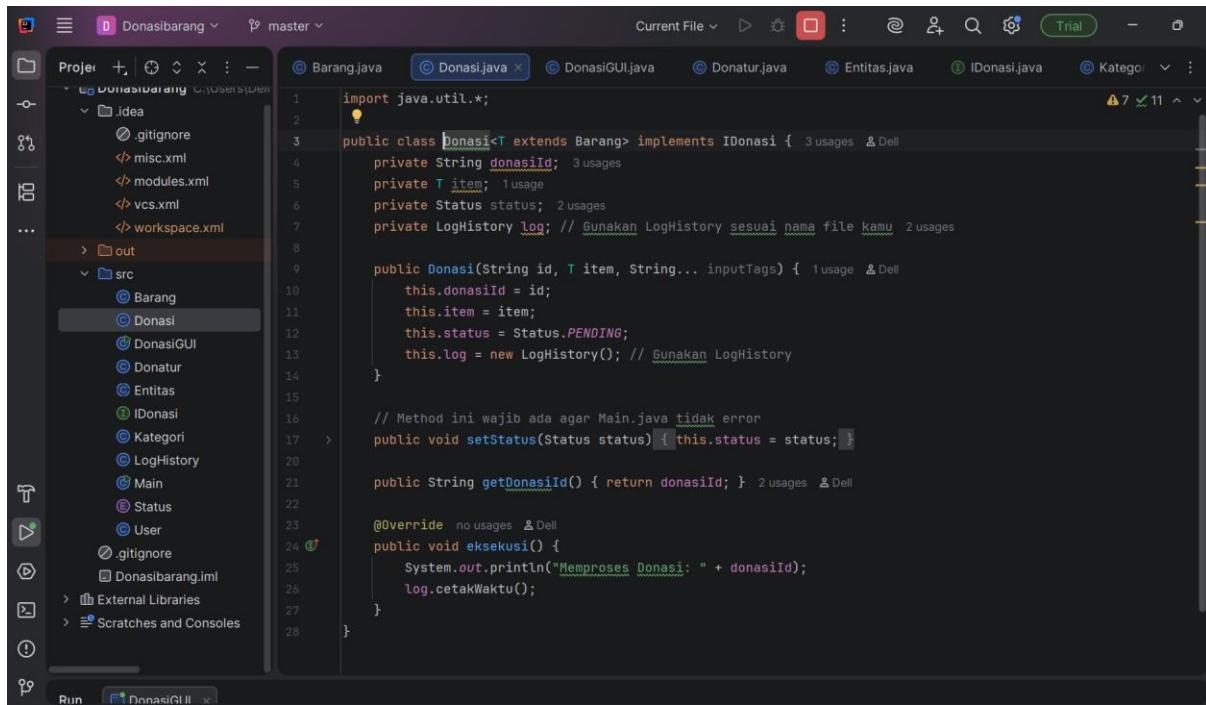
### 3. Class Barang

The screenshot shows the IntelliJ IDEA interface with the project 'Donasibarang' open. The left sidebar displays the project structure with packages like .idea, .gitignore, misc.xml, modules.xml, vcs.xml, workspace.xml, out, and src containing classes such as Barang, Donasi, DonasiGUI, Donatur, Entitas, IDonasi, Kategori, LogHistory, Main, Status, and User. The current file is 'Barang.java', which defines a class 'Barang'. The code includes methods for adding attributes and setting/getting image paths.

```
1 import java.util.*;
2
3 public class Barang { 6 usages & Dell
4     private String nama; 2 usages
5     private Kategori kategori; 2 usages
6     private Map<String, String> atribut = new HashMap<>(); 2 usages
7     private String imagePath; // Tambahan untuk gambar 2 usages
8
9     public Barang(String nama, Kategori kategori) { 2 usages & Dell
10        this.nama = nama;
11        this.kategori = kategori;
12    }
13
14    public void tambahAtribut(String kunci, String nilai) { this.atribut.put(kunci, nilai); }
15
16    // Getter dan Setter untuk Gambar
17    public void setImagePath(String path) { this.imagePath = path; } no usages & Dell
18    public String getImagePath() { return imagePath; } no usages & Dell
19    public String getNama() { return name; } 1 usage & Dell
20    public Kategori getKategori() { return kategori; } no usages & Dell
21    public Map<String, String> getAtribut() { return atribut; } 3 usages & Dell
22
23    }
24}
```

Digunakan untuk menyimpan data barang yang didonasikan.

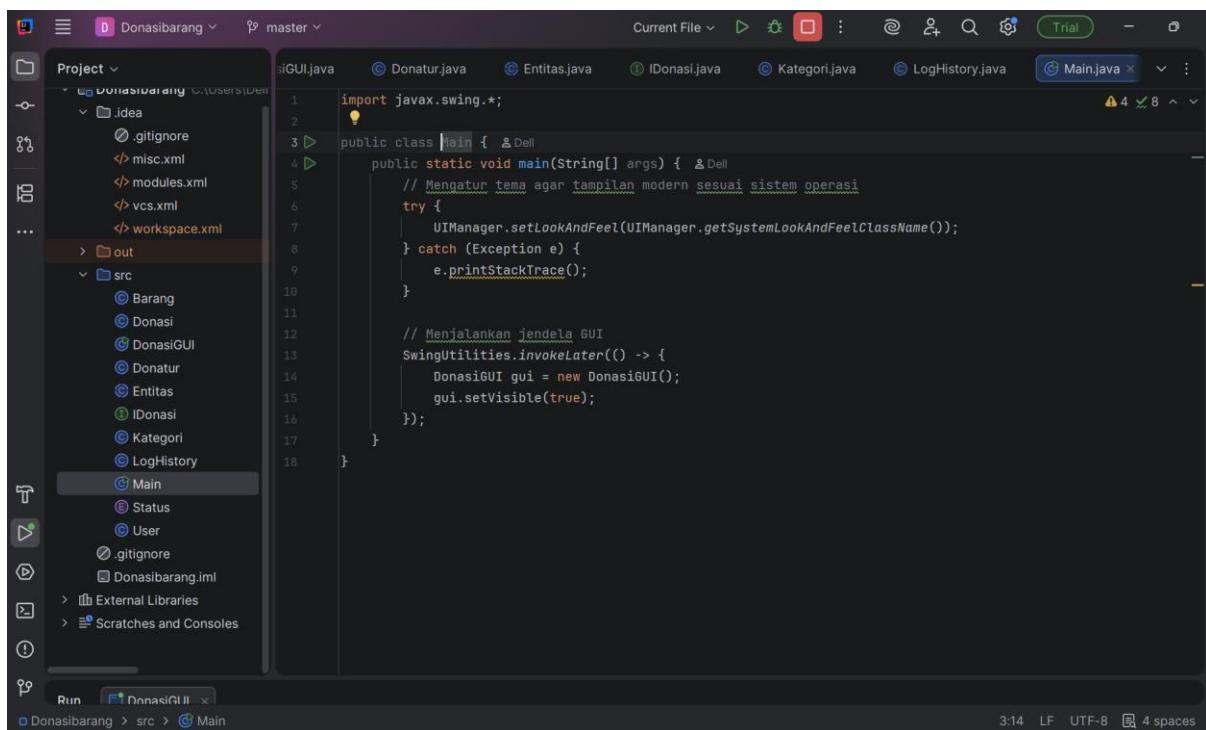
### 4. Class Donasi



```
import java.util.*;  
public class Donasi<T extends Barang> implements IDonasi {  
    private String donasiId;  
    private T item;  
    private Status status;  
    private LogHistory log; // Gunakan LogHistory sesuai nama file kamu  
  
    public Donasi(String id, T item, String... inputTags) {  
        this.donasiId = id;  
        this.item = item;  
        this.status = Status.PENDING;  
        this.log = new LogHistory(); // Gunakan LogHistory  
    }  
  
    // Method ini wajib ada agar Main.java tidak error  
    public void setStatus(Status status) { this.status = status; }  
  
    public String getDonasiId() { return donasiId; }  
  
    @Override no usages & Dell  
    public void eksekusi() {  
        System.out.println("Memproses Donasi: " + donasiId);  
        log.cetakWaktu();  
    }  
}
```

Mengelola proses donasi, status donasi, serta relasi antar objek.

## 5. Class MainApp



```
import javax.swing.*;  
public class Main {  
    public static void main(String[] args) {  
        // Mengatur tema agar tampilan modern sesuai sistem operasi  
        try {  
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
  
        // Menjalankan jendela GUI  
        SwingUtilities.invokeLater(() -> {  
            DonasiGUI gui = new DonasiGUI();  
            gui.setVisible(true);  
        });  
    }  
}
```

Berfungsi sebagai entry point aplikasi dan mengatur alur program secara keseluruhan.

(Sertakan potongan kode program dari repository)

---

## G. Penjelasan Screenshot Tampilan Aplikasi

Screenshot menampilkan:

### 1. Form Input Donasi (Bagian Kiri)

The screenshot shows a window titled "Sistem Aplikasi Donasi Barang Layak Pakai". On the left side, there is a form with four input fields: "Nama Donatur", "Kategori Barang", "Nama Barang", and "Ukuran". Below these fields are two buttons: "Tambah Detail Spesifik" (grey background) and "Kirim Donasi" (green background). On the right side, there is a table with columns: ID, Donatur, Barang (Kateg...), Detail, Tags, and Status. The table currently has no data.

- **Nama Donatur**  
Digunakan untuk memasukkan nama orang yang melakukan donasi barang.
- **Kategori Barang**  
Digunakan untuk menentukan kategori barang yang didonasikan, seperti pakaian, buku, atau perlengkapan rumah tangga.
- **Nama Barang**  
Digunakan untuk memasukkan nama barang yang akan didonasikan.
- **Ukuran**  
Digunakan untuk mengisi ukuran barang (contoh: S, M, L, atau ukuran lainnya sesuai barang).
- **Tombol “Tambah Detail Spesifik”**  
Berfungsi untuk menambahkan informasi tambahan terkait barang donasi.
- **Tombol “Kirim Donasi”**  
Digunakan untuk mengirim data donasi agar diproses dan ditampilkan pada tabel.

### 2. Tabel Data Donasi (Bagian Kanan)

ID	Donatur	Barang (Kateg...)	Detail	Tags	Status

**Tabel ini digunakan untuk menampilkan data donasi yang telah dimasukkan, dengan kolom:**

- **ID : Nomor unik donasi**
- **Donatur : Nama donatur**
- **Barang (Kategori) : Nama dan kategori barang**
- **Detail : Informasi tambahan barang**
- **Tags : Label atau penanda barang**
- **Status : Status donasi (misalnya: *Menunggu, Diproses, Terkirim*)**

### 3. Judul aplikasi

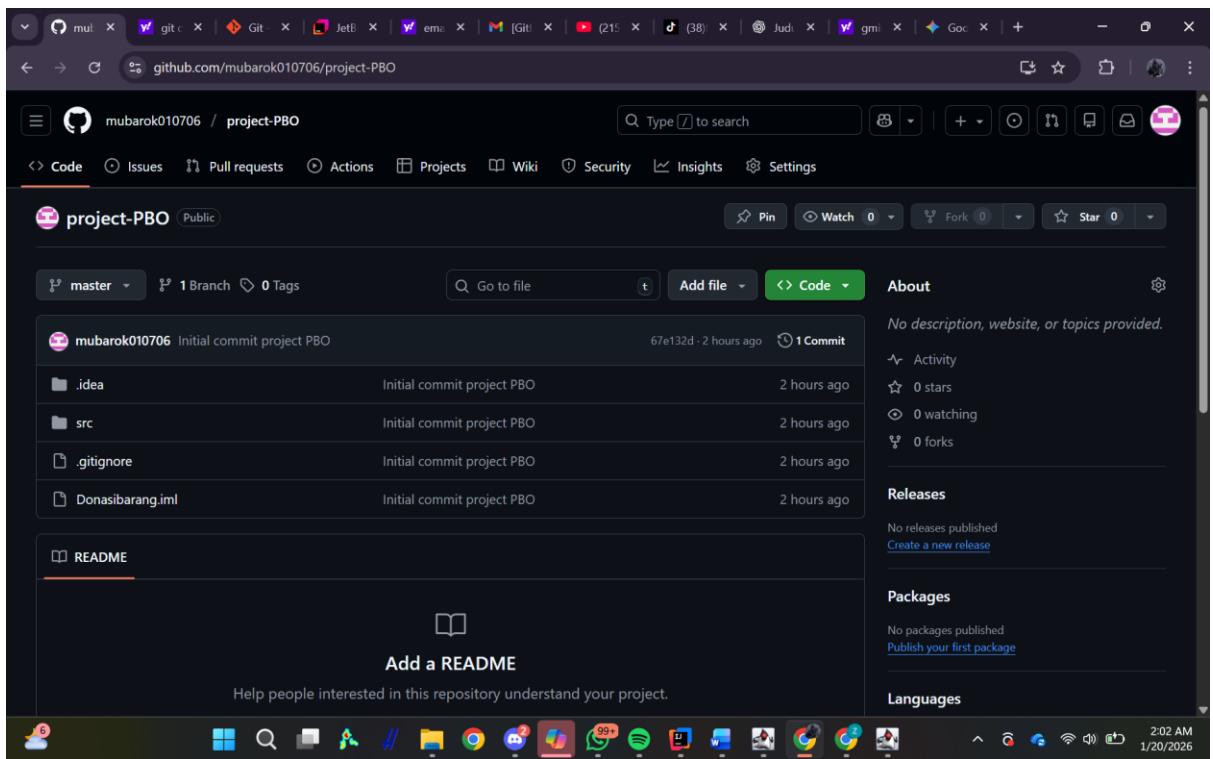


**“Sistem Aplikasi Donasi Barang Layak Pakai” ditampilkan pada bagian atas jendela sebagai identitas aplikasi.**

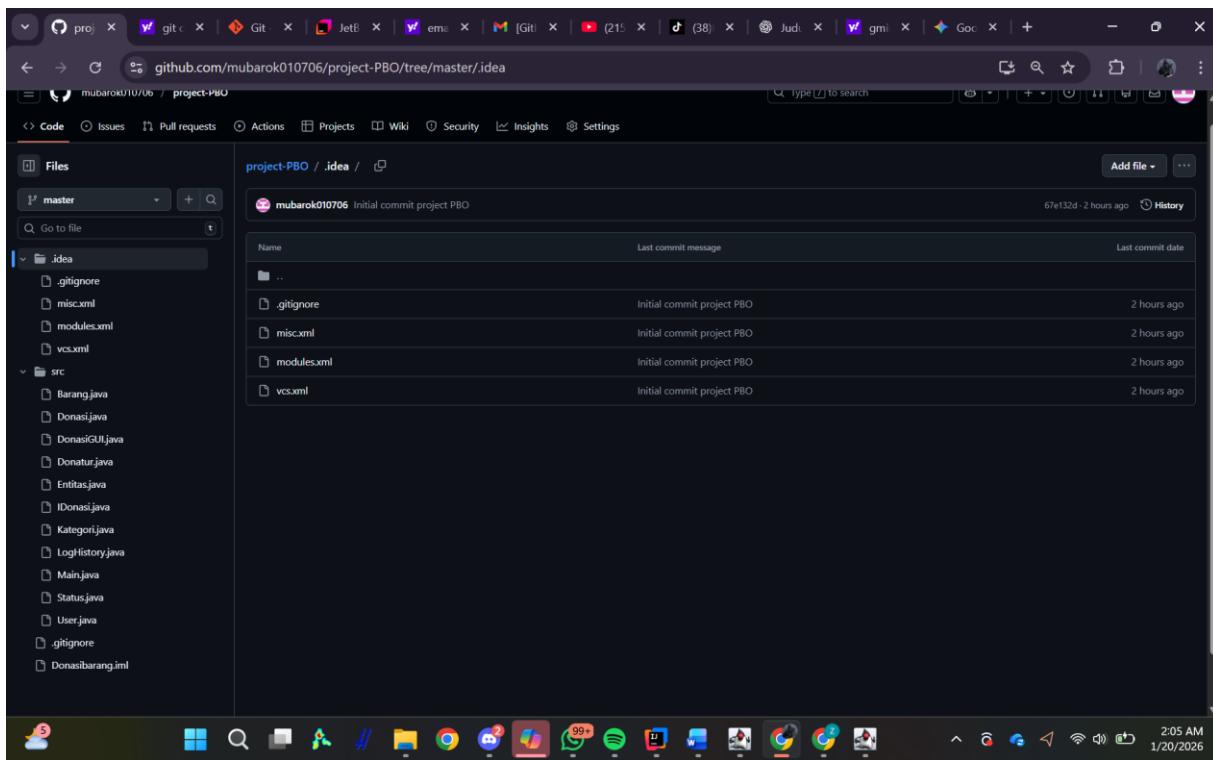
### H. Screenshot Status Unggah di GitHub/GitLab

**Bagian ini menampilkan:**

#### 1. Struktur folder project



## 2. File source code Java



## I. Analisis Pengerjaan Proyek

### 1. Waktu Pengerjaan

**Proyek dikerjakan selama kurang lebih X hari, dimulai dari analisis kebutuhan hingga implementasi dan pengujian.**

## **2. Ketercapaian Spesifikasi**

**Sebagian besar spesifikasi berhasil diimplementasikan sesuai perencanaan, terutama penerapan konsep OOP.**

## **3. Biaya yang Dibutuhkan**

**Pengembangan aplikasi ini tidak memerlukan biaya finansial karena menggunakan perangkat lunak open source.**

## **4. Kendala**

- **Pemahaman awal konsep OOP**
- **Penyesuaian relasi antar class**

## **5. Tantangan dan Pengembangan ke Depan**

- **Pengembangan aplikasi berbasis web atau mobile**
- **Integrasi database**
- **Penambahan fitur notifikasi dan laporan donasi**

---

## **Kesimpulan**

**Aplikasi Donasi Barang Layak Pakai Berbasis OOP berhasil dikembangkan dengan menerapkan konsep-konsep utama pemrograman berorientasi objek dan memenuhi kriteria penilaian UAS PBO.**