**LAPORAN AKHIR TUGAS BESAR  
PRAKTIKUM PEMOGRAMAN 1**

**INVENTARIS ALAT KANTOR**



**Dipersiapkan Oleh:**

**Kelas:** A

|  |  |
| --- | --- |
| **Ketua:** | 233040097 - Danny Lukman |
| **Anggota:** | 233040018 - Muhammad Haekal Al Ghazy |
|  | 233040129 - Aria Yudhistira |
|  | 233040019 - Fauzi Ahmad Ramdani |
|  |  |

**Nama Asisten Laboratorium:**

Dzikri Setiawan, Muhammad Daffa Musyaffa, Muhamad Marsa Nur Jaman

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PASUNDAN BANDUNG**

**2024**

# Daftar Isi

[**Daftar Isi 2**](#_iz7cp713lh53)

[**1. Pendahuluan 3**](#_twrl604ypu4t)

[1.1. Deskripsi 3](#_a9f6g2nqswg8)

[1.2. Lingkup Pengerjaan 3](#_qjnn9gnbkdk9)

[1.3. Dokumentasi Pengerjaan 3](#_jkv2t2are0xs)

[1.4. Pembagian Tugas 3](#_7jj39fdhp5g)

[**2. Analisis dan Perancangan 4**](#_k0a74da9fd7p)

[2.1. Analisis Kebutuhan Sistem 4](#_wvxnzx7oia2m)

[2.1.1. Kebutuhan Fungsional 4](#_g1ukzaqdlhzh)

[2.1.2. Kebutuhan Non Fungsional 4](#_sgxqrh1amxlf)

[**3. Implementasi Perangkat Lunak 5**](#_qnt3fpbx2j9v)

[3.1. Kakas dan Perangkat Lunak 5](#_fg08jspqamy1)

[3.1.1. Spesifikasi Kakas/Hardware 5](#_mhmr821wnavk)

[3.1.2. Daftar Perangkat Lunak 5](#_2d6412mxwicp)

[3.2. Struktur Proyek 5](#_yustu4957nxy)

[3.3. Implementasi Kelas dan Struktur Data 6](#_gspd3mph5205)

[3.4. Fitur-Fitur Aplikasi 6](#_bol664fuwjww)

[**4. Kesimpulan 7**](#_68fwnsohu3aq)

[**Lampiran 8**](#_6f78wy6dgsp2)

# Pendahuluan

## Deskripsi

Aplikasi **Inventaris Alat Kantor** adalah sistem manajemen inventaris yang dirancang untuk mempermudah pengelolaan alat kantor dengan menggunakan struktur data ***Queue*** (Antrean). Sistem ini memungkinkan pengguna untuk menambahkan alat ke dalam antrean, memproses alat pertama yang dimasukkan, serta menampilkan alat yang ada dalam antrean sesuai dengan prinsip ***FIFO*** (*First In First Out*). Aplikasi ini bertujuan untuk membantu dalam pengelolaan alat kantor secara efisien dan terstruktur.

## Lingkup Pengerjaan

**Lingkup pengerjaan tugas besar ini mencakup:**

* Pengembangan aplikasi berbasis console untuk mengelola inventaris alat kantor.

**Fitur utama yang dikembangkan adalah:**

* Menambah alat baru ke dalam antrean.
* Memproses alat pertama yang ada di antrean (FIFO).
* Menampilkan alat yang ada dalam antrean yang belum diproses.

**Batasan Sistem:**

* Setiap alat yang dimasukan ke antrean akan diproses sesuai dengan urutan kedatangan.
* Pengguna akan menginput alat secara manual melalui konsol

## Dokumentasi Pengerjaan

Proyek ini disimpan di repository GitHub dengan akses publik. Anda dapat mengakses kode sumber dan dokumentasi melalui link berikut:.

<https://github.com/danny221004/Ngabring_id_A_PP1>

Panduan Akses Repository:

1. Buka Link Repository

* Akses URL repository di atas melalui browser Anda.

1. Clone Repository

* Klik tombol Code (berwarna hijau) di halaman repository.
* Pilih opsi HTTPS dan salin URL yang muncul.

1. Buka terminal atau Git Bash di komputer Anda dan jalankan perintah berikut:

## Pembagian Tugas

Berikut adalah pembagian tugas pengerjaan dalam proyek ini:

| **NPM** | **Nama** | **Tugas yang Dikerjakan** |
| --- | --- | --- |
| 233040097 | Danny Lukman | Membuat kelas Node dan Queue |
| 233040129 | Aria Yudhistira | Mengimplementasikan InventarisAlatKantor |
| 233040018 | Muhammad Haekal Al Ghazy | Mengimplementasikan InventarisAlatService |
| 233040019 | Fauzi Ahmad Ramdani | Mengembangkan Main dan Pengujian |

# Analisis dan Perancangan

## Analisis Kebutuhan Sistem

### Pada bagian ini, dilakukan identifikasi terhadap kebutuhan yang diperlukan dalam pengembangan sistem. Fokuskan pada fitur-fitur utama dan dukungan teknis yang dibutuhkan. Berikut adalah kebutuhan yang terstruktur:

### Kebutuhan Fungsional

**CRUD Data dengan Prinsip FIFO (First In, First Out):**

Aplikasi harus menyediakan fitur untuk mengelola data dalam antrian menggunakan prinsip FIFO. Pengguna dapat melakukan operasi berikut:

* + *Create* (Tambah): Menambahkan elemen baru ke dalam antrian.
  + *Read* (Lihat): Menampilkan elemen-elemen yang ada dalam antrian.
  + *Update* (Ubah): Mengubah data yang ada dalam antrian jika diperlukan.
  + *Delete* (Hapus): Menghapus elemen dari antrian, dengan elemen pertama yang masuk menjadi elemen pertama yang keluar.

**Validasi Input:**

* Input yang diberikan oleh pengguna harus divalidasi untuk memastikan konsistensi dan format yang benar.

**Antrian:**

* Aplikasi harus mendukung operasi antrian dengan prinsip FIFO, yaitu menambahkan elemen di belakang dan menghapus elemen dari depan secara efisien.

### Kebutuhan Non Fungsional

**Kecepatan Respon Aplikasi:**

* Aplikasi harus memberikan respon yang cepat terhadap interaksi pengguna.
* Interface yang Mudah Digunakan (Console-Based):
* Antarmuka pengguna harus dirancang secara sederhana dan intuitif.

**Validasi Input yang Memadai:**

* Setiap input yang diberikan oleh pengguna harus diperiksa untuk memastikan bahwa data yang dimasukkan sesuai dengan format yang diinginkan.

**Error Handling yang Baik:**

* Aplikasi harus memiliki mekanisme penanganan kesalahan yang efektif.

**Efisiensi Penggunaan Memori:**

* Aplikasi harus dioptimalkan agar menggunakan memori secara efisien.
  1. **Perancangan Struktur Data**

Struktur data utama yang digunakan dalam aplikasi ini adalah ***Queue* (Antrian)**, yang diimplementasikan untuk menangani pemrosesan data secara berurutan dengan prinsip ***FIFO (First In, First Out)****.* Berikut adalah cara struktur data ini dirancang dan diimplementasikan dalam aplikasi:

### **Desain Struktur Data Queue**

Dalam aplikasi ini, ***Queue*** dirancang untuk menyimpan data secara urut dan memprosesnya sesuai dengan prinsip *FIFO*. Data pertama yang dimasukkan ke dalam antrian akan menjadi yang pertama kali diproses dan dikeluarkan.

* ***Enqueue*:** Menambahkan elemen ke belakang antrian.
* ***Dequeue*:** Mengambil dan menghapus elemen dari depan antrian.
* ***Front*:** Melihat elemen yang ada di depan antrian tanpa menghapusnya.
* ***IsEmpt*y:** Memeriksa apakah antrian kosong atau tidak.
* ***Size:*** Mengetahui jumlah elemen yang ada dalam antrian.
  + 1. **Alasan Pemilihan Struktur Data Queue**

*Queue* dipilih karena prinsip *FIFO* sangat sesuai dengan aplikasi yang memerlukan pemrosesan data secara berurutan, misalnya:

* **Sistem Antrian Pelanggan:** Dalam aplikasi layanan pelanggan, setiap permintaan atau pelanggan yang datang pertama kali harus dilayani lebih dahulu.
* **Manajemen Tugas atau Proses:** Sistem operasi atau aplikasi lain yang memproses tugas secara berurutan, di mana tugas yang pertama kali diterima harus diproses terlebih dahulu.
* **Pengolahan Data Streaming:** Misalnya, dalam pemrosesan data yang datang bertahap, aplikasi memerlukan *buffer* atau antrian untuk menyimpan dan mengolah data sesuai urutan.
  + 1. **Cara Kerja Struktur Data Queue dalam Aplikasi**
* ***Enqueue* (Menambah Elemen):**  
  Ketika data baru ditambahkan ke dalam antrian (misalnya permintaan pelanggan atau tugas baru), operasi ***Enqueue*** akan menambahkannya di bagian belakang antrian. Elemen ini akan menunggu untuk diproses setelah elemen-elemen sebelumnya.
* ***Dequeue* (Mengambil Elemen):**  
  Ketika aplikasi memproses data (misalnya memproses permintaan pelanggan), operasi ***Dequeue*** akan menghapus dan mengeluarkan elemen dari bagian depan antrian. Elemen yang dikeluarkan adalah elemen yang pertama kali masuk ke dalam antrian *(FIFO).*
* ***Front* (Melihat Elemen Depan):**  
  Pengguna atau aplikasi dapat melihat elemen yang berada di depan antrian tanpa menghapusnya. Hal ini berguna untuk mengetahui tugas atau data mana yang akan diproses berikutnya.
* ***IsEmpty* (Memeriksa Kekosongan Antrian):**  
  Aplikasi dapat memeriksa apakah antrian kosong atau tidak dengan menggunakan fungsi***isEmpty***. Jika antrian kosong, maka aplikasi dapat melakukan tindakan tertentu seperti menunggu data baru atau memberi pemberitahuan kepada pengguna.
* ***Size* (Mengetahui Ukuran Antrian):**  
  Dengan menggunakan fungsi ***size***, aplikasi dapat mengetahui berapa banyak elemen yang ada dalam antrian. Ini berguna untuk memantau berapa banyak tugas atau permintaan yang sedang menunggu untuk diproses.
  1. **Perancangan Algoritma**

Pada aplikasi Inventaris Alat Kantor, terdapat beberapa operasi dasar yang menggunakan algoritma *FIFO* (*First In First Out*) untuk mengelola antrean alat kantor.

**1. Algoritma *Enqueue* (Menambah Alat ke Antrean)**

* Menambahkan alat baru ke antrean:
* Terima input alat (nama, jumlah, status).
* Tambahkan alat ke antrean.

**2. Algoritma *Dequeue* (Mengambil dan Memproses Alat)**

Mengambil alat pertama dan memprosesnya:

* Periksa antrean.
* Ambil alat pertama, ubah statusnya, dan proses alat tersebut.

**3. Algoritma *Display* (Menampilkan Alat yang Belum Diproses)**

Menampilkan semua alat dalam antrean:

* Periksa apakah antrean kosong.
* Tampilkan semua alat yang belum diproses.

# Implementasi Perangkat Lunak

## Kakas dan Perangkat Lunak

### **Spesifikasi Kakas/Hardware**

Prosesor: Intel i7

RAM: 24GB

Penyimpanan: 500GB SSD

Sistem Operasi: Windows 10

### **Daftar Perangkat Lunak**

IDE: IntelliJ IDEA

Java Version: JDK 24

Version Control: Git, GitHub

## Struktur Proyek

Organisasi folder dalam proyek ini adalah sebagai berikut:

*Ngabring\_id\_A\_PP1/*

*├── src/*

*│ ├── entity/ # Kelas data model (Alat)*

*│ ├── services/ # Logika bisnis (InventarisAlatService, InventarisAlatKantor)*

*│ └── Main.java # Kelas utama aplikasi*

*├── docs/ # Dokumentasi*

*└── README.md # Panduan proyek*

## Implementasi Kelas dan Struktur Data

Kelas yang Diimplementasikan:

| **No.** | **Nama File** | **Package** | **Deskripsi Fungsi** |
| --- | --- | --- | --- |
| 1 | Alat.java | entity | Kelas untuk merepresentasikan alat kantor |
| 3 | InventarisAlatService.java | services | Kelas untuk mengelola layanan terkait alat kantor. |
| 4 | InventarisAlatKantor.java | services | Kelas untuk pengelolaan antrean alat kantor. |
| 5 | Main.java |  | Kelas utama yang menyediakan antarmuka pengguna berbasis console. |

**Penjelasan Struktur Data:**

**1. Alat.java**

Fungsi: Kelas ini digunakan untuk merepresentasikan alat kantor.

**Atribut:**

* nama: Nama alat.
* jumlah: Jumlah alat.
* status: Status alat (misalnya "Belum diproses", "Telah diproses").

**Metode Utama:**

* *getNama(), getJumlah(), getStatus()*: Untuk mendapatkan nilai atribut.
* *setStatus()*: Untuk mengubah status alat.
* *toString():* Untuk menampilkan informasi alat.

**2. InventarisAlatKantor.java**

Fungsi: Kelas ini mengelola antrean alat kantor.

Struktur Data: Menggunakan *Queue* (Antrean) untuk menyimpan objek Alat.

**Metode Utama:**

* *enqueue*(Alat alat): Menambahkan alat baru ke dalam antrean.
* *dequeue*(): Mengambil dan memproses alat pertama dalam antrean.
* *displayAla*t(): Menampilkan semua alat yang belum diproses.

**3. InventarisAlatService.java**

Fungsi: Kelas ini bertugas untuk mengelola operasional antrean alat kantor.

**Metode Utama:**

* *enqueue*(Alat alat): Memanggil metode *enqueue*() di kelas InventarisAlatKantor untuk menambah alat.
* *dequeue*(): Memanggil metode *dequeue*() di kelas InventarisAlatKantor untuk memproses alat.
* *displayAla*t():Memanggil metode *displayAlat*() di kelas InventarisAlatKantor untuk menampilkan alat.

**4. Main.java**

Fungsi: Kelas ini berfungsi sebagai antarmuka pengguna berbasis console.

**Fitur:**

* Menampilkan menu interaktif kepada pengguna untuk mengelola alat (menambah alat, memproses alat, menampilkan alat).
* Mengambil input dari pengguna dan menjalankan operasi yang sesuai menggunakan InventarisAlatService.

## Fitur-Fitur Aplikasi

Berikut adalah penjelasan untuk beberapa fitur yang telah dikembangkan dalam aplikasi **Inventaris Alat Kantor**

**Fitur 1: Tambah Alat Baru**

**Deskripsi:**  
Fitur ini memungkinkan pengguna untuk menambahkan alat baru ke dalam antrean. Pengguna akan diminta untuk memasukkan nama alat dan jumlah alat, kemudian alat tersebut akan dimasukkan ke dalam antrean untuk diproses lebih lanjut.

**Cara Kerja:**

* 1. Pengguna memilih menu "Tambah Alat Baru".
  2. Sistem meminta pengguna untuk memasukkan nama alat.
  3. Sistem meminta pengguna untuk memasukkan jumlah alat.
  4. Sistem membuat objek Alat dan menambahkan alat tersebut ke dalam antrean menggunakan metode *enqueue*().
  5. Pesan konfirmasi ditampilkan bahwa alat berhasil ditambahkan.

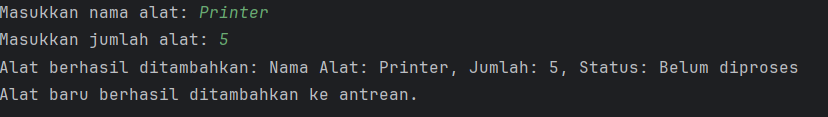
**Input:**

* Nama alat (misalnya: "Printer").
* Jumlah alat (misalnya: "5").

**Output:**

* Pesan yang menunjukkan bahwa alat telah berhasil ditambahkan ke antrean.

**Screenshot:**



**Fitur 2: Proses Alat yang Ada**

**Deskripsi:**

Fitur ini memungkinkan pengguna untuk memproses alat pertama yang ada di antrean. Alat pertama yang masuk akan diproses dan statusnya akan diubah menjadi "Telah diproses".

**Cara Kerja:**

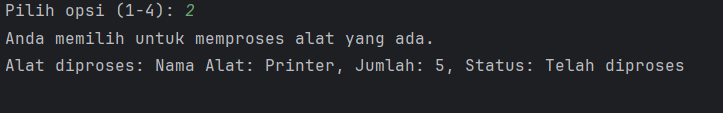
* Pengguna memilih menu "Proses Alat yang Ada".
* Sistem memeriksa apakah antrean alat kosong.
* Jika ada alat, sistem mengambil alat pertama dari antrean menggunakan metode dequeue().
* Sistem mengubah status alat menjadi "Telah diproses".
* Sistem menampilkan alat yang telah diproses.

**Input:**

* Tidak ada input yang diperlukan selain memilih opsi ini dari menu.

**Output:**

* Pesan yang menampilkan alat yang telah diproses dan mengubah statusnya.

**Screenshot:**

**Fitur 3: Tampilkan Daftar Alat yang Belum Diproses**

Fitur ini menampilkan semua alat yang ada dalam antrean yang belum diproses. Ini memungkinkan pengguna untuk melihat daftar alat yang harus diproses.

**Cara Kerja:**

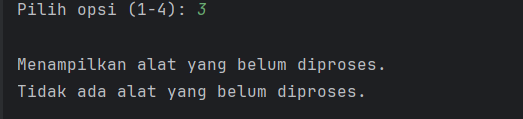
* Pengguna memilih menu "Tampilkan Daftar Alat yang Belum Diproses".
* Sistem memeriksa antrean dan menampilkan semua alat yang ada di antrean yang belum diproses.
* Jika antrean kosong, sistem akan menampilkan pesan bahwa tidak ada alat yang perlu diproses.

**Input:**

* Tidak ada input yang diperlukan selain memilih opsi ini dari menu.

**Output:**

* Daftar alat yang belum diproses.
* Jika antrean kosong, pesan yang menyatakan bahwa tidak ada alat yang perlu diproses.

**Screenshot:**

**Fitur 4: Keluar dari Aplikasi**

Fitur ini memungkinkan pengguna untuk keluar dari aplikasi dan menghentikan program.

**Cara Kerja:**

* Pengguna memilih menu "Keluar".
* Aplikasi akan berhenti dengan menampilkan pesan terima kasih.
* Program akan menutup aplikasi.

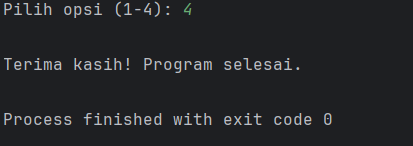
**Input:**

* Tidak ada input yang diperlukan selain memilih opsi keluar.

**Output:**

* Pesan terima kasih dan aplikasi akan berhenti.

**Screenshot:**



# Kesimpulan

**Pencapaian:**

Fitur-fitur yang berhasil diimplementasikan:

* Tambah Alat Baru: Pengguna dapat menambah alat ke dalam antrean.
* Proses Alat yang Ada: Alat pertama yang masuk diproses dan statusnya diubah.
* Tampilkan Daftar Alat yang Belum Diproses: Menampilkan semua alat yang belum diproses dalam antrean.
* Keluar: Pengguna dapat keluar dari aplikasi dengan aman.

**Struktur data yang berhasil diterapkan:**

* *Queue* (Antrian): Struktur data Queue berhasil diterapkan untuk mengelola alat dengan prinsip *FIFO (First In First Out).* Antrean alat dikelola dengan menggunakan *LinkedLis*t, memungkinkan penambahan dan penghapusan alat dengan efisien.

**Pembelajaran yang diperoleh:**

* Pemahaman *Queue*: Pembelajaran tentang struktur data antrian dan cara mengelolanya dengan benar.
* Penggunaan Java: Memperdalam penggunaan struktur data Java seperti *LinkedList* dan konsep-konsep dasar Pratikum Pemrograman.

**Tantangan yang Dihadapi:**

Kesulitan teknis yang ditemui:

* Pengelolaan validasi *input* agar aplikasi menerima data yang sesuai dan menghindari kesalahan input oleh pengguna.
* Mengimplementasikan antrean menggunakan *Queue* dan memahami cara kerjanya dalam konteks pengelolaan alat kantor.

**Cara mengatasi masalah tersebut:**

* *Validasi Input*: Menambahkan mekanisme validasi untuk memastikan *input* yang diterima sesuai dengan format yang diinginkan.
* Pemahaman Antrean: Memperdalam pemahaman mengenai struktur data *Queue* dan cara implementasinya dalam aplikasi.

**Evaluasi:**

Kelebihan dan kekurangan sistem:

**Kelebihan:**

* Sistem efisien dan mudah digunakan.
* Implementasi struktur data *Queue* berhasil diterapkan dengan prinsip *FIFO*, memastikan alat diproses sesuai urutan.

**Kekurangan:**

* Antarmuka aplikasi hanya berbasis konsol, yang mungkin kurang *user-friendly* bagi beberapa pengguna.
* Tidak ada fitur pengelolaan lebih lanjut untuk alat yang sudah diproses, seperti penyimpanan data atau laporan.

**Potensi pengembangan lebih lanjut:**

* Mengembangkan antarmuka grafis (GUI) untuk membuat aplikasi lebih *user-friendly.*
* Menambahkan fitur pencarian dan pengurutan alat berdasarkan kategori atau status.

**Rekomendasi:**

Saran untuk pengembangan atau perbaikan selanjutnya:

* Menambahkan GUI untuk mempermudah pengguna dalam mengoperasikan aplikasi.
* Menambahkan fitur untuk mengelola alat yang telah diproses, seperti penyimpanan data alat yang telah diproses dalam file atau database.

**Fitur tambahan yang bisa diimplementasikan:**

* Fitur Edit Alat: Untuk memperbarui informasi alat (nama, jumlah, status).
* Pencarian Alat: Fitur untuk mencari alat berdasarkan nama atau status.
* Laporan: Fitur untuk menghasilkan laporan alat yang diproses dan yang belum diproses.

Dengan kesimpulan ini, aplikasi Inventaris Alat Kantor telah berhasil mengimplementasikan fitur dasar untuk mengelola alat dengan struktur data *Queue* dan prinsip *FIFO*. Pengembangan lebih lanjut dapat dilakukan untuk memperluas fungsionalitas dan meningkatkan pengalaman pengguna.

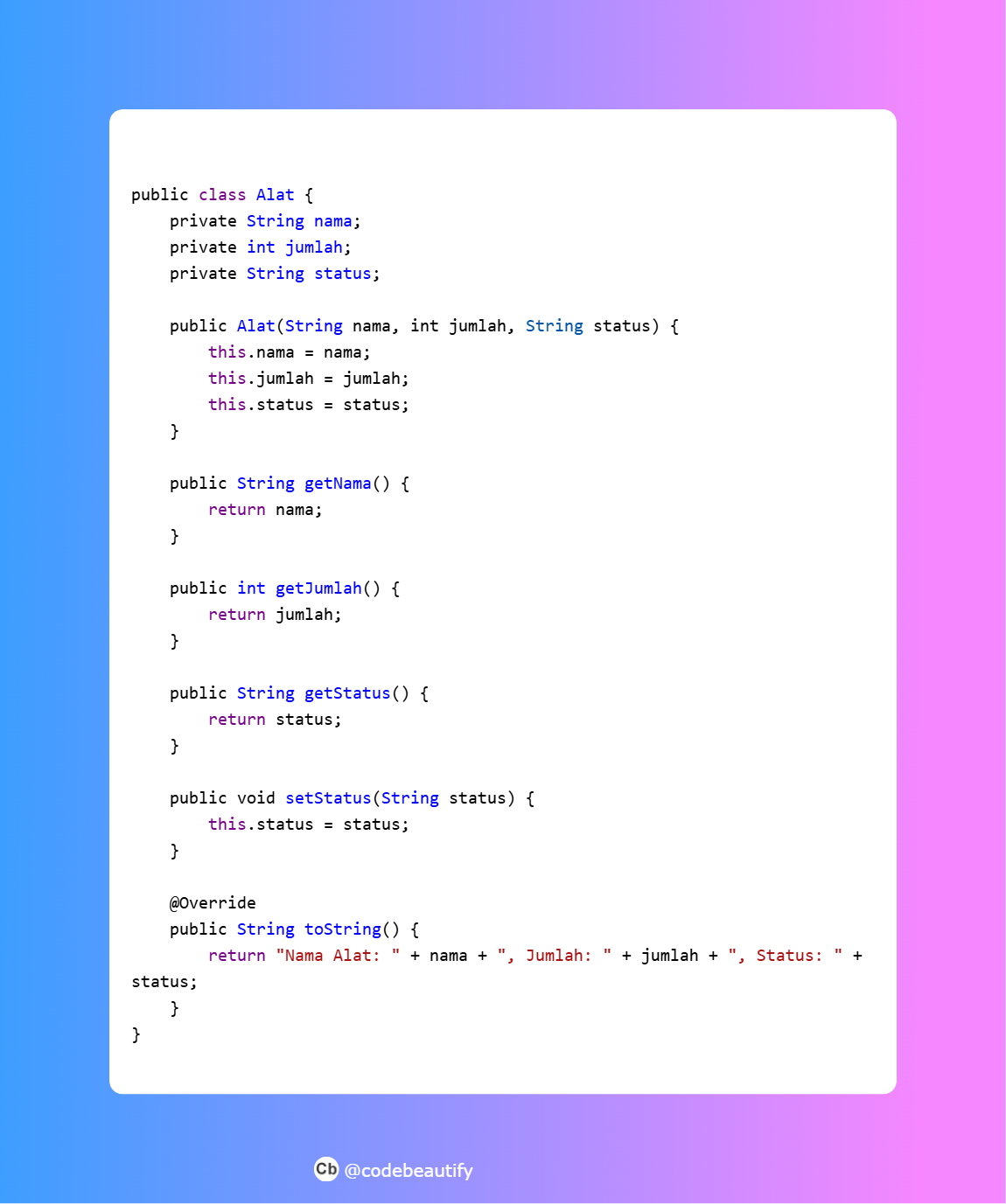
# Lampiran

Lampiran

A. Kode Sumber

Berikut adalah potongan kode penting dari setiap kelas yang telah diimplementasikan dalam aplikasi Inventaris Alat Kantor:

**Alat.java**

Kelas ini digunakan untuk merepresentasikan objek alat yang terdiri dari nama, jumlah, dan status alat.

**InventarisAlatKantor.java**  
Kelas ini mengelola antrean alat menggunakan **Queue** dan prinsip FIFO.



**InventarisAlatService.java**  
Kelas ini mengelola operasional antrean alat menggunakan metode yang disediakan oleh **InventarisAlatKantor**.

**Main.java**  
Kelas utama yang menyediakan antarmuka pengguna berbasis console.



B. Repository GitHub

Link lengkap ke repository GitHub:

<https://github.com/danny221004/Ngabring_id_A_PP1>

Panduan instalasi dan menjalankan aplikasi:

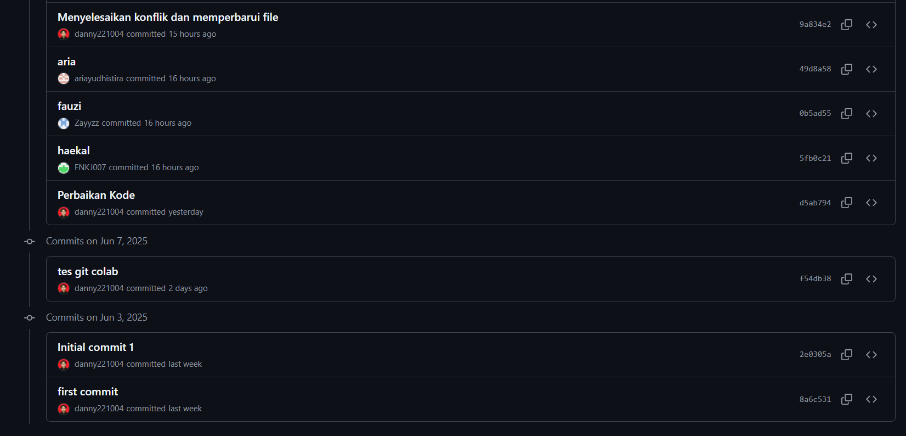
Clone Repository:

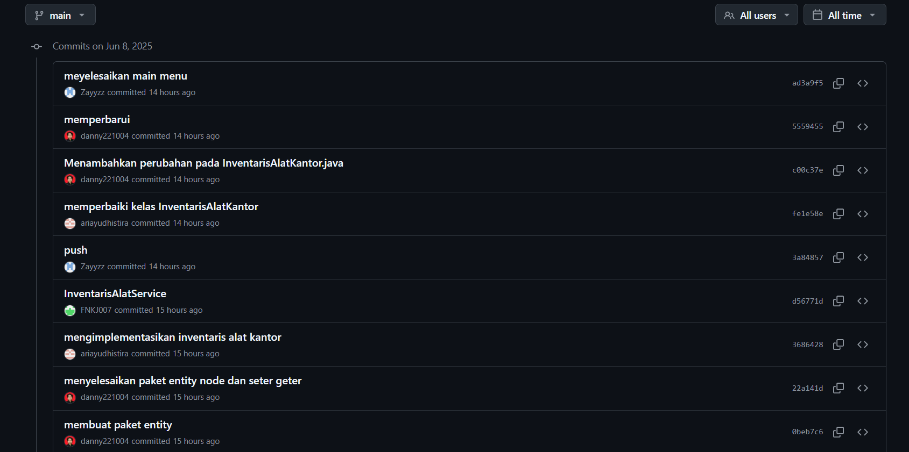
* Gunakan perintah berikut di terminal:

Jalankan Program:

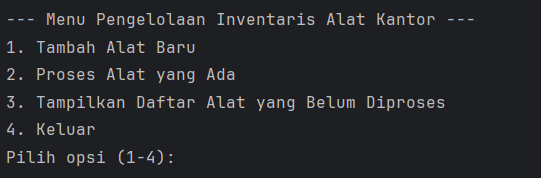
* Buka file **Main.java** dan jalankan menggunakan IDE (misalnya IntelliJ IDEA atau Eclipse) atau melalui terminal:

C. Log Aktivitas GitHub



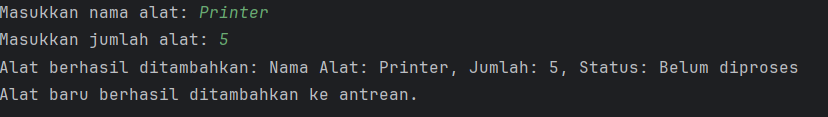


D. Screenshot Aplikasi

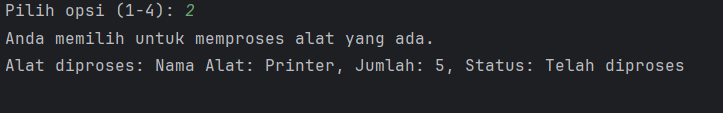
Tampilan Menu Utama:

**Output dari Setiap Fitur:**

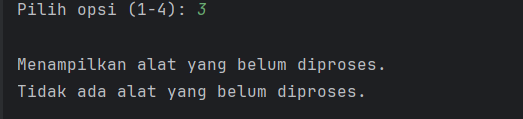
Fitur Tambah Alat:



Fitur Proses Alat:



Fitur Tampilkan Alat:



Fitur Tampilkan Keluar:

