## **🧭 Metodologi Penelitian dan Pengembangan**

### **1. Metode Pengembangan**

Metodologi yang digunakan adalah **Metode Waterfall** karena aplikasi memiliki alur pengembangan linier dan bertahap, dimulai dari perencanaan hingga implementasi dan pengujian.

**Tahapan:**

1. **Analisis Kebutuhan**
   * Studi pustaka seputar produksi pakaian dan algoritma greedy
   * Identifikasi parameter produksi: jenis kain, ukuran, kebutuhan kain per ukuran, dll.
2. **Desain Sistem**
   * Perancangan antarmuka (UI) dengan Tkinter
   * Desain arsitektur modular (main.py, logic.py, ui.py)
   * Desain algoritma greedy + optimasi sisa kain
3. **Implementasi**
   * Pembuatan kode Python
   * Integrasi data JSON sebagai dataset fleksibel
   * Visualisasi hasil dengan Matplotlib
4. **Pengujian**
   * Validasi input pengguna
   * Pengujian akurasi perhitungan dan visualisasi hasil
   * Uji coba distribusi kain dan optimasi sisa
5. **Evaluasi**
   * Bandingkan hasil aplikasi dengan perhitungan manual
   * Uji kenyamanan UI dan respons aplikasi

## **⚙️ Desain Sistem**

### **1. Arsitektur Modular**

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

| main.py | <-- Entry point, menginisialisasi UI

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

|

v

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

| ui.py |<---->| logic.py |

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

|

v

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

| data/jenispakaian.json |

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

### **2. Flowchart Sistem**

Berikut adalah flowchart utama proses penghitungan produksi optimal:

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

| Mulai (main.py) |

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

|

v

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

| Input: Jenis produk, |

| jenis kain, total kain |

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

|

v

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

| Pilih ukuran & persentase |

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

|

v

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

| Klik "Hitung Produksi" |

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

|

v

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

| Hitung alokasi awal (Greedy Algorithm) |

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

|

v

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

| Optimasi sisa kain (optional)|

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

|

v

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

| Tampilkan hasil (tabel + chart) |

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

|

v

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

| Selesai |

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

## **🧮 Algoritma yang Digunakan**

### **📌 *Greedy Algorithm*:**

* Alokasikan kain sesuai persentase fokus pengguna.
* Prioritaskan produksi sebanyak mungkin per ukuran.
* Tidak mempertimbangkan kombinasi semua kemungkinan (non-ekshaustif).

### **📌 *Optimasi Sisa Kain (Optional)*:**

* Hitung sisa kain setelah alokasi awal.
* Cek apakah cukup untuk membuat satu pakaian dari ukuran termurah.
* Tambahkan ke ukuran tersebut hingga kain habis atau tidak cukup.

## **🔍 Validasi dan Pengujian**

**Jenis Pengujian:**

* **Functional Testing**: Validasi semua tombol, input, dan output.
* **Boundary Testing**: Uji batas kain minimum, maksimum, dan edge-case persentase.
* **Logic Testing**: Bandingkan hasil distribusi manual dengan sistem.
* **GUI Testing**: Uji tata letak, responsivitas, dan konsistensi antarmuka.

## **📊 Hasil yang Diharapkan**

* Pengguna dapat mengetahui jumlah produksi maksimal dari ketersediaan kain
* Efisiensi distribusi kain sesuai ukuran yang diinginkan
* Minimasi limbah/sisa kain
* Visualisasi yang membantu pemahaman data produksi