# 📄 Report Generator Documentation

# Created by: Nico Ardian Nugroho

# Year: 2025

This application is a GUI-based utility built with Python and ttkbootstrap to generate Word reports from structured Excel data across multiple branches. It supports multithreaded processing, error logging, and uses Word templates for dynamic content injection.

## 🎯 Features

- GUI interface with modern themes (ttkbootstrap)  
- Data extraction from specific sheets in Excel files  
- Word template (.docx) rendering using docxtpl  
- Live progress bar and status messages  
- Preview panel for logs and data details  
- Thread-safe execution for better UI responsiveness  
- Output logs saved in Excel format  
- Toast notifications for feedback

## 🧩 Key Components

### 1. show\_toast(title, message, duration)

Displays a toast notification using ttkbootstrap's ToastNotification.

### 2. extract\_data\_from\_excel(file\_path, source\_key)

Extracts predefined cells from two sheets: 'Result Data' and 'Simulasi Transaksi'.  
Returns a dictionary with prefixed keys based on the source.

### 3. generate\_report(template\_path, output\_path, context)

Fills a Word template (.docx) using docxtpl and saves it as a new report.

### 4. process\_folders()

Main function for iterating through branch directories, reading Excel files,  
collecting values, rendering reports, and updating the GUI preview panel.  
Handles missing files and logs all actions.

### 5. threaded\_processing()

Runs the processing in a background thread to avoid freezing the GUI.

### 6. save\_log\_to\_excel(log\_data, folder)

Stores the processing results for each branch in a log Excel file.

### 7. center\_window(app, width, height)

Utility to center the application window on the screen.

## 🖼️ GUI Overview

- Entry for report maker name  
- Folder selection for parent directory  
- File selection for Word template  
- Progress bar for visual feedback  
- Preview panel with color-coded logs  
- Footer with author info

## 📁 File Output

- One Word document per branch  
- A generation\_log.xlsx file with process results

## ✅ Status Tag Colors

- Green: Success  
- Orange: Missing files  
- Red: Errors  
- Blue: Informational  
- Teal: Extracted data  
- Purple: Branch header

## 📌 Notes

- Excel files must contain the expected sheets and cell structure.  
- Word template must use {{ }} Jinja-style variables matching the keys from `context`.  
- Ensure all libraries are installed: openpyxl, docxtpl, ttkbootstrap

---  
© 2025 Created by SOW 7