📘 Data Processing Application Documentation

# Created by: Nico Ardian Nugroho

# Year: 2025

# 1. Overview

The 'Processing Data Mission Digital' is a Python GUI application designed using ttkbootstrap and openpyxl.   
It enables users to process Excel files ('combined.xlsx') found in nested folders by calculating digital performance statistics,   
appending calculated sheets, and saving results in 'result.xlsx'. The interface includes user identity input, real-time progress tracking,   
and detailed status logs with cancel capability.

# 2. Features

- GUI built with ttkbootstrap (light/dark mode toggle)  
- Selects parent folder containing Excel files  
- Reads and processes 'combined.xlsx' in all subfolders  
- Adds calculated sheets: Calculated Data, Result Data, Simulasi Transaksi  
- Saves outputs to 'result.xlsx'  
- Logs process with detailed status updates  
- Shows real-time progress and elapsed time  
- Supports cancellation and auto-close prompt on completion

# 3. GUI Components

- Nama and NIP input fields  
- Folder selection and Start button  
- Dark Mode toggle  
- Progress bar  
- Status labels (status, elapsed time)  
- Scrolled log box for real-time logging  
- Cancel button

# 4. Processing Logic

The core logic begins with folder selection and input validation. The app recursively walks through all subfolders,  
checks for 'combined.xlsx', and if found, calls `calculate\_and\_save\_excel` to process the file and save 'result.xlsx'.  
If not found, the folder is skipped. Results are logged, and progress is displayed in real-time.

# 5. Excel Processing

The `calculate\_and\_save\_excel` method reads the active worksheet of the workbook, computes digital percentages,   
adds styling and border formatting, creates calculated summaries and simulation data for future projections,   
and saves all to a new sheet in the same workbook under a new filename.

# 6. Output Sheets

- 'Calculated Data': Includes Periode, Mesin, Counter, Total, Digital % (with red fill if < 85%)  
- 'Result Data': Contains average % and transaction needs to meet 85%  
- 'Simulasi Transaksi': Projects target daily transactions over various day intervals

# 7. Logging & Feedback

Each action is logged to both the UI and a log file saved in the selected folder. The progress bar reflects completion percent.   
After processing, a toast appears, followed by an optional close prompt.

# 8. Cancellation Handling

Clicking the '⛔ Cancel' button sets a flag that interrupts the processing loop safely. This ensures no partial file corruption.

# 9. Dependencies

- Python Standard Library: os, math, time, datetime, threading, tkinter  
- Third-party: openpyxl, ttkbootstrap

# 10. Suggested Enhancements

- Add file preview before processing  
- Enable processing only selected folders  
- Support multilingual interface  
- Export summary to a separate report (PDF or Excel)