Overview

The "File Processor" is a Python-based application designed to read data from multiple files located in an input directory, process the data to remove duplicates, compute additional information and output the results to a CSV file in a specified output directory. The output CSV includes a footer with the second-highest and average salaries.

Implementation Details

- **Language**: Python 3.x
- **Libraries**: `csv`, `os`, `unittest`
- **Script Execution**:
    - `run.py` for processing files.
    - `run_tests.py` for running unit tests.

Data Flow

1. **Input**:
    - Files are read from the `/input` directory.
    - Data is parsed and loaded into memory.
2. **Processing**:
    - Data is processed to remove duplicates.
    - `Gross Salary` is computed for each record.
3. **Output**:
    - Data is written to `/output/RESULT_NEW.csv`.
    - Footer with the second highest and average `Gross Salary` is appended.

Code Structure

- **src/file_processor.py class FileProcessor**:
    - `read_files_from_folder(folder_path)`: Reads and combines data from files in the specified folder.
    - `process_data(data)`: Removes duplicates and processes the data.
    - `calculate_footer(salaries)`: Computes the second highest and average `Gross Salary`.
    - `write_to_csv(output_file_path, data, footer)`: Writes processed data and footer to a CSV file.
- **tests/test_file_processor.py**:
    - Unit tests for each function in `file_processor.py`.

Design Considerations

1. **Modularity**:
    - Each function in `file_processor.py` should handle a specific part of the processing pipeline, ensuring that the code is modular and easier to test.
2. **Scalability**:
    - The design should allow for easy addition of new data processing steps or output formats if needed in the future.