A Hybrid Portfolio Optimization and Rebalancing Tool

Born from my deep personal interest and continuous independent research into financial markets, this project was created to address a critical challenge for any investor: **how to allocate capital across a portfolio of diverse assets optimally.** My experience has shown that selecting good individual assets is only half the battle; structuring them into a resilient and efficient portfolio is paramount. My goal was to build a comprehensive, interactive program that guides an investor through the entire portfolio management workflow—from screening candidate assets to optimizing allocations based on their risk profile, and finally, projecting the portfolio's future growth.

A Structured Approach to Portfolio Management

This project was implemented as a complete, end-to-end command-line tool in Python. It is designed to walk a user through a structured, four-step process: (1) Asset Screening, (2) Portfolio Optimization, (3) Rebalancing Action Plan, and (4) Probabilistic Forecasting. The architecture is built on a specific design philosophy aimed at creating a tool that is not only quantitatively robust but also practical and flexible for real-world use.

## I. Methodological Rationale & Design Philosophy

Before building the tool, I conducted research to select methodologies that would address the practical needs of a modern investor and overcome the limitations of simpler models. The following design choices form the foundation of this project:

- **Why a Hybrid Data Model?** My primary consideration was flexibility. I designed the tool to accept both **automated data via Yahoo Finance** and **manual user input** for key metrics (CAGR, Volatility). This allows for a truly holistic portfolio analysis that reflects an investor's entire net worth, not just their stock holdings.
- **Why Pre-Screen Assets by Risk Profile?** Before optimization, the tool screens and ranks candidate assets based on the user's stated risk profile. I implemented this step to align the asset selection with the investor's core philosophy from the outset. By filtering candidates based on **Volatility (for Conservative)**, **Sharpe Ratio (for Moderate)**, or **CAGR (for Aggressive)**, the tool ensures that the final optimized portfolio is constructed from assets that are already philosophically aligned with the user's goals.
- **Why Resampled Efficient Frontier?** For the core optimization, I chose the **Resampled Efficient Frontier** method over classic Mean-Variance Optimization (MVO). Resampling combats the instability of classic MVO by running hundreds of simulations and averaging the results, producing portfolios that are **more robust, stable, and diversified**.
- **Why Monte Carlo Simulation for Forecasting?** I added the **Monte Carlo simulation** as the final step to translate the characteristics of the optimized portfolio into a dynamic, forward-looking probabilistic forecast. It answers the crucial question: "Given this optimal asset mix, what is the probable range of my portfolio's value in the future?"

II. The Asset Allocation Engine: From Optimization to Actionable Choice

This is the analytical core of the program. After running the Resampled Efficient Frontier optimization, the tool does not simply provide one answer; it generates a spectrum of three distinct, actionable portfolio allocations. This is designed to give the user a clear understanding of the risk-reward trade-offs:

1. **The Konservatif Portfolio:** This portfolio is designed for capital preservation and stability. It prioritizes minimizing volatility by allocating a larger portion of the portfolio to historically less volatile assets.
2. **The Agresif Portfolio:** This portfolio represents a maximum growth strategy. It concentrates capital into the asset that has demonstrated the highest historical growth potential, accepting higher individual asset risk in pursuit of the highest possible returns.
3. **The Moderat Portfolio:** This is the most balanced recommendation, created through the core resampling optimization process. It is mathematically structured to find an efficient "sweet spot" on the risk-return spectrum, seeking to deliver the most effective return for the level of risk undertaken.
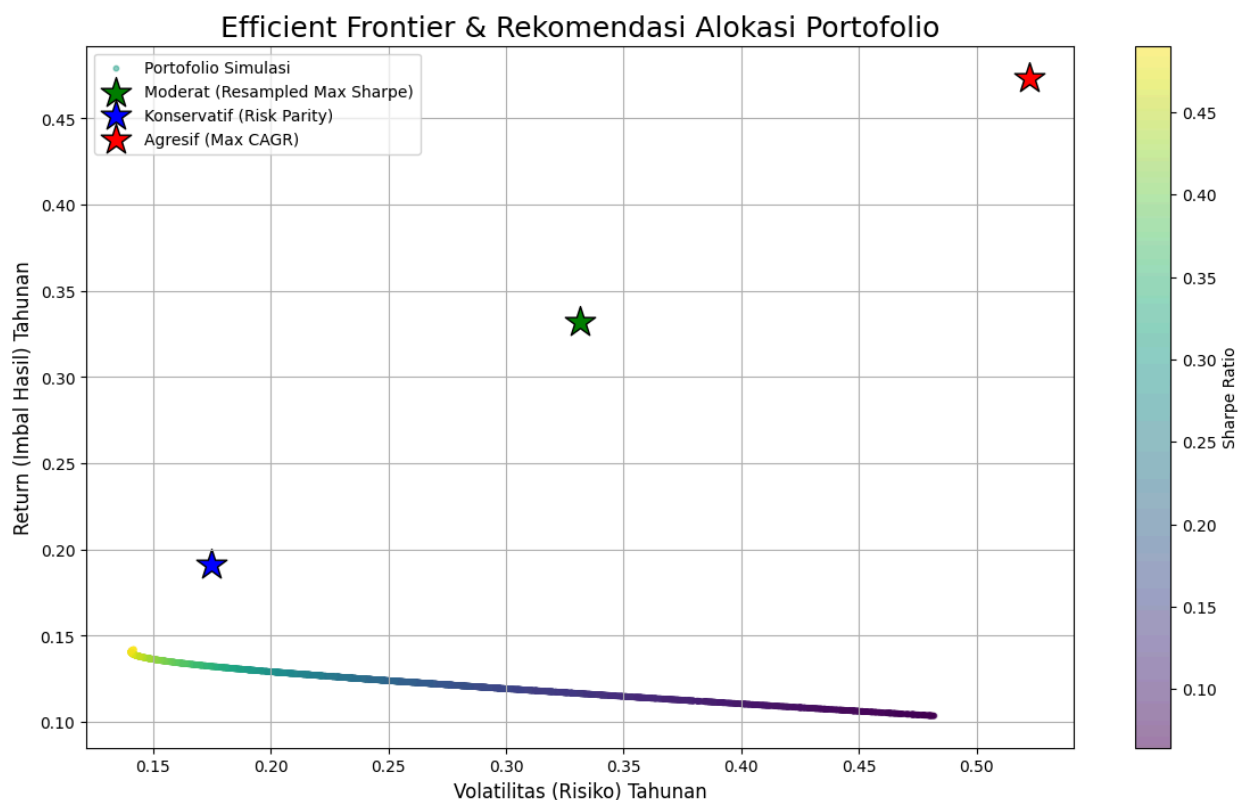
Efficient Frontier & Portfolio Recommendations: A chart visualizing thousands of simulated portfolio allocations, with the three recommended portfolios (Conservative, Moderate, Aggressive) highlighted as stars.

III. Implementation & Detailed Case Study Results

To demonstrate the tool's practical application, I ran a case study with the following setup: an existing portfolio of Rp 1,800,000 in **Asset A**, Rp 3,000,000 in new capital, a **"Moderat"** risk profile, and a universe of 8 candidate assets.

- **Asset Screening & Optimization:** The tool first selected **Asset A** (Sharpe Ratio: 1.06) and **Asset B** (Sharpe Ratio: 0.57) as the top candidates. It then generated the three optimal portfolios as described above.
- **Rebalancing Plan:** As the chosen profile was "Moderat," the program generated a clear action plan to align the holdings with the 60.84% / 39.16% target. It advised the user to **"TAMBAH/BELI" (Add/Buy) Rp 1,120,437** worth of Asset A and **"TAMBAH/BELI" Rp 1,879,563** worth of the newly added Asset B.

Asset	Hist. CAGR	Current Value	Target Value	Action	Action Amount	Reason
Asset A	62.43%	Rp 1,800,000	Rp 2,920,437	ADD/BUY	Rp 1,120,437	Allocate to optimal weight of 60.84%.
Asset B	15.72%	Rp 0	Rp 1,879,563	ADD/BUY	Rp 1,879,563	Allocate to optimal weight of 39.16%.

IV. Forward-Looking Analysis: Monte Carlo Simulation of the Optimal Portfolio

After defining the optimal "Moderat" portfolio, the tool's final step was to run a forward-looking stress test using a **15,000-iteration Monte Carlo simulation**. This simulation provided a rich, probabilistic view of what the investor could expect from this specific allocation. The 5-year forecast was particularly insightful:

- In a **Median (P50) scenario**, the portfolio was projected to achieve a **nominal growth of +201.32%**. Even after accounting for a high 9% annual inflation rate, the expected **real growth was an impressive +95.83%**.
- In a **Pessimistic (P5) scenario**, the portfolio's nominal value was projected to decline by only **-10.67%** over the full five years, providing a clear measure of downside risk.
- Most importantly, the simulation calculated a **93.23% probability of the portfolio generating a profit** over the 5-year period. This single, intuitive metric provides a powerful confidence level for the chosen investment strategy.

Simulasi Monte Carlo - Proyeksi Nilai Portofolio (5 Tahun)
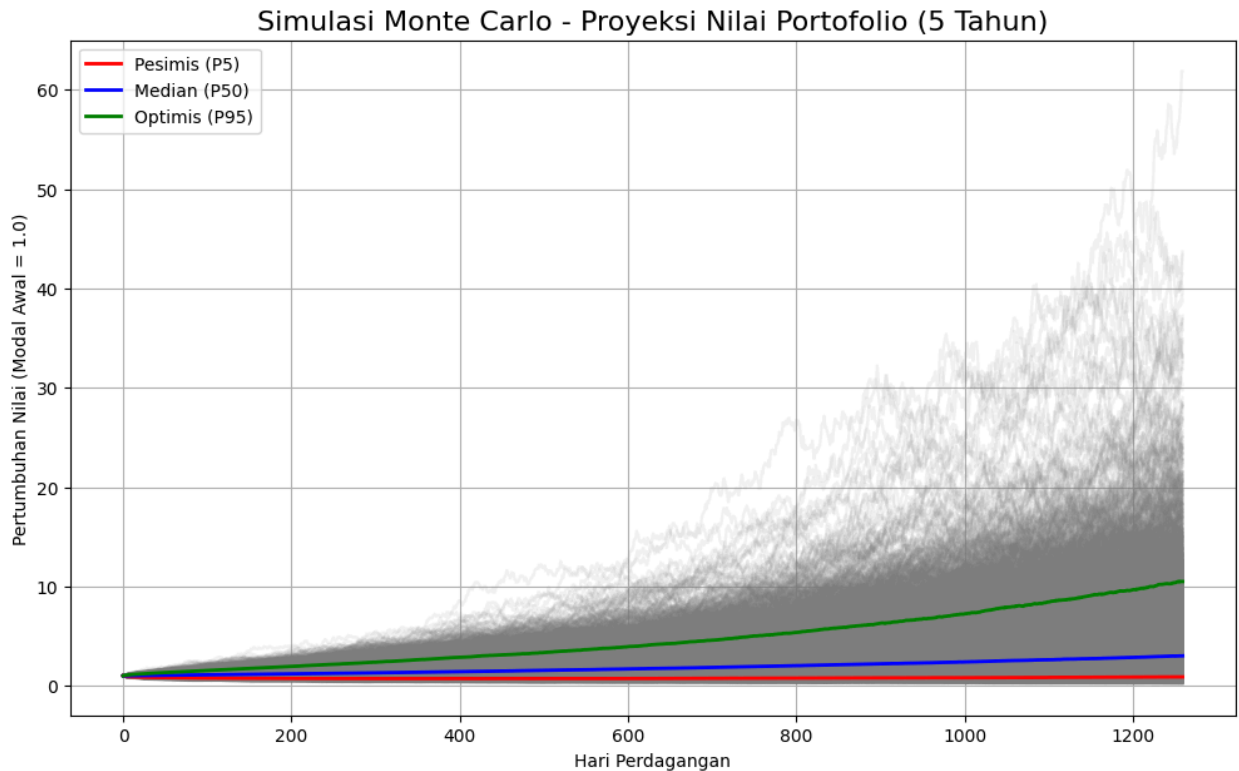
Monte Carlo Simulation of the Optimized Portfolio: A visualization showing the range of potential growth outcomes for the user's selected portfolio over various time horizons.

Project Conclusion

This project successfully integrates several advanced financial theories into a single, user-friendly, and practical tool. It guides an investor from a disorganized list of candidate assets to a statistically robust, optimized portfolio tailored to their risk profile, complete with a clear rebalancing plan and a probabilistic forecast of its future performance. By deliberately choosing methodologies like the Resampled Efficient Frontier and a hybrid data model, the tool provides a sophisticated yet accessible solution to the fundamental challenge of portfolio allocation and management.

Skills & Competencies Demonstrated

- **Portfolio Theory & Optimization:** Implementing Modern Portfolio Theory (MPT) and the more advanced Resampled Efficient Frontier technique.
- **Financial Modeling:** Creating a complete, multi-step financial workflow.
- **Risk Profile Analysis:** Designing a system that tailors its output based on user-defined risk profiles.
- **Probabilistic Forecasting:** Applying Monte Carlo simulations to model the future performance of a multi-asset portfolio.

- **Python for Finance:** Using libraries like `pandas`, `numpy`, and `yfinance` for complex financial calculations and data handling.
- **Data Visualization:** Creating informative charts like the Efficient Frontier and Monte Carlo projections using `matplotlib`.