Automating Fundamental Analysis: A Data-Driven Approach to Stock Ranking Using Singular Value Decomposition

Daniel Anindito Nugroho - 13524002
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13524002@std.stei.itb.ac.id, danielnugroho900@gmail.com

***Abstract*—Fundamental analysis is a cornerstone of investment decision-making, requiring the evaluation of numerous financial metrics to determine a company's intrinsic value. However, manually analyzing multidimensional financial data for hundreds of stocks is cognitively demanding, time-consuming, and prone to subjective bias. This study proposes a mathematical approach to automate stock ranking using Singular Value Decomposition (SVD). By representing stocks as vectors in a high-dimensional Euclidean space defined by key valuation, profitability, and risk metrics, this method utilizes SVD to extract latent quality factors from the data matrix. This approach offers a strong, data-driven alternative to traditional screening methods by objectively identifying fundamentally strong companies without human intervention. The implementation, applied to constituents of the Indonesia Stock Exchange (IDX), demonstrates the method's capability to effectively rank stocks based on intrinsic quality while maintaining computational efficiency. The results suggest that linear algebra techniques can serve as a powerful tool for automated financial analysis and portfolio selection.**

***Keywords*—SVD, Fundamental Analysis, Stock Ranking, Linear Algebra, Vector Space Model, Indonesia Stock Exchange.**

I. Introduction

In the realm of capital markets, rational investment decisions rely heavily on fundamental analysis. This method evaluates a company's financial health by examining various ratios to determine its intrinsic value [6]. However, the Indonesia Stock Exchange (IDX) comprises hundreds of listed issuers, and analyzing them manually is a complex task. The primary challenge investors face is the curse of dimensionality. Analyzing multiple financial ratios simultaneously for a large number of stocks exceeds human cognitive capacity, often leading to subjective bias and inefficiency.

This study aims to solve this problem by utilizing a dataset of 194 liquid stocks listed on the IDX. To ensure a comprehensive evaluation, we incorporate 12 distinct fundamental metrics covering valuation, profitability, and risk. These metrics are Price-to-Earnings Ratio (PE Ratio), Price-to-Book Value (PBV), Price-to-Sales Ratio (PS Ratio), EV/EBITDA, Return on Equity (ROE), Return on Assets (ROA), Gross Margin, Operating Margin, Debt-to-Equity Ratio (DER), Current Ratio, Quick Ratio, and Free Cash Flow (FCF). Processing this high-dimensional data requires an objective computational method to simplify the complexity into a single actionable ranking.

This problem can be modeled using concepts from Linear Algebra. Each of the 194 stocks can be conceptualized as a vector within a 12-dimensional Euclidean space ($\mathbb{R}^{12}$), where each dimension represents one of the aforementioned financial metrics [2], [3]. By representing financial data as a set of vectors, matrix operations can be applied to extract latent patterns from the dataset effectively.

This paper proposes the application of Singular Value Decomposition (SVD) to automate stock ranking. SVD is a matrix factorization technique capable of reducing data dimensionality while preserving its most critical information [4], [5]. In this context, SVD is employed to project the complex 12-dimensional data matrix into a principal dimension representing the latent quality of the companies.

This research applies theories studied in the Linear Algebra and Geometry course to real-world cases in the Indonesian stock market. The implementation is conducted using the Python programming language, utilizing the yfinance library to retrieve market data [7]. The results are expected to demonstrate that a vector-based mathematical approach can provide stock recommendations that are objective, efficient, and fundamentally strong.

II. THEORETICAL FOUNDATION

To automate the stock ranking process effectively, it is essential to establish a strong theoretical framework that bridges financial fundamental analysis with linear algebra concepts. This section details the financial metrics selected, the market segmentation used, and the mathematical models applied.

A. Fundamental Analysis and Financial Metrics

A stock represents a fractional ownership in a corporation. Fundamental analysis is the method of evaluating a security's intrinsic value by examining related economic and financial factors. The ultimate goal is to arrive at a number that an investor can compare with a security's current price to see whether the security is undervalued or overvalued [6].

In this study, the multidimensional quality of a company is defined by 12 quantitative metrics. These are categorized into three distinct dimensions to ensure a holistic evaluation of the firm's health:

1) Valuation: These metrics assess whether a stock is expensive or cheap relative to its financial performance.

- Price-to-Earnings Ratio (PE Ratio): Defined as price divided by EPS (Earnings per Share). It measures how much investors are willing to pay per dollar of earnings. A lower PE ratio typically indicates a cheaper valuation, though it may also reflect low growth expectations.
- Price-to-Book Value (PBV): Defined as price divided by book value per share. It compares the market valuation to the company's net asset value. A value < 1 suggests the stock is trading below its liquidation value, often sought by value investors.
- Price-to-Sales Ratio (PS Ratio): Defined as market cap divided by revenue. This metric is crucial for evaluating companies with thin margins or net losses (such as tech startups), as revenue is harder to manipulate than net income.
- EV/EBITDA: The ratio of Enterprise Value to Earnings Before Interest, Taxes, Depreciation, and Amortization. Unlike P/E, this metric is neutral to capital structure (debt vs. equity) and taxation, making it ideal for comparing capital-intensive industries across different tax jurisdictions.

2) Profitability: These metrics measure management's ability to generate earnings from the company's resources.

- Return on Equity (ROE): Defined as net income divided by shareholders equity. It measures the profitability relative to shareholder capital. A high ROE indicates a company has a competitive advantage (economic moat).
- Return on Assets (ROA): Defined as net income divided by total assets. It indicates how efficient management is at using its assets to generate earnings, regardless of financing structure.
- Gross Profit Margin: Defined as (revenue - COGS) divided by revenue. High gross margins indicate strong pricing power and brand equity, allowing the company to withstand cost inflation.
- Operating Margin: Defined as operating income divided by revenue. It reflects operational efficiency after accounting for variable costs but before interest and taxes.

3) Risk and Liquidity: These metrics evaluate the company's solvency and ability to meet obligations.

- Debt-to-Equity Ratio (DER): Defined as total liabilities divided by equity. High DER indicates high financial leverage. While leverage can amplify returns, it significantly increases bankruptcy risk during economic downturns.
- Current Ratio: Defined as current assests divided by current liabilities. Measures the ability to pay short-term obligations due within one year. A ratio below 1.0 suggests potential liquidity problems.
- Quick Ratio: Defined as (current assets - inventory) divided by current liabilities. A stringent test of liquidity that excludes inventory, which may be illiquid or difficult to sell quickly during a crisis.
- Free Cash Flow (FCF): Operating Cash Flow minus Capital Expenditures. Positive FCF indicates the company generates surplus cash to pay dividends, reduce debt, or reinvest in growth without relying on external financing.

B. Stock Market Segmentation and Data Scope

This study utilizes a purposive sampling method to ensure the results are representative of the Indonesian economy. The dataset comprises 194 stocks derived from major indices on the Indonesia Stock Exchange (IDX):

1) LQ45 Index (The Blue Chips): This segment consists of the 45 most liquid stocks with the largest market capitalization. These companies represent the "Blue Chips" of Indonesia, typically characterized by mature business models, consistent dividends, and high stability.

2) Kompas100 Index: A broader index covering 100 stocks, including the LQ45 constituents plus 55 other companies with strong fundamentals and high liquidity. This segment introduces "Second-Liner" stocks that often offer a balance between stability and higher growth potential compared to the giant caps.

3) Pefindo25 Index: An index curated by PEFINDO (Credit Rating Agency) focusing on small-to-medium enterprises (SMEs) with excellent financial performance and good administrative governance. Inclusion of this segment allows the SVD model to identify high-quality undervalued stocks ("Hidden Gems") often overlooked by institutional investors.

By filtering these indices to exclude companies with incomplete historical data (such as recent IPOs), the final dataset of 194 stocks captures over 80% of the total market capitalization of the IDX.

C. Vector Space Representation

In the domain of Linear Algebra, data objects can be modeled as vectors in a high-dimensional space. A single stock i is represented as a vector $\mathbf{v}_i$ in a Euclidean space $\mathbb{R}^{12}$ [2], [3]:

$$\mathbf{v}_i = \begin{bmatrix} m_{i,1} & m_{i,2} & \dots & m_{i,12} \end{bmatrix}^T \tag{1}$$

Where $m_{i,j}$ corresponds to the j-th financial metric of stock i. Consequently, the entire market dataset of 194 stocks is represented as a matrix $A \in \mathbb{R}^{194 \times 12}$.

Raw financial data is heterogeneous in scale and direction. To construct a valid vector space, three steps are performed:

1) Data Filtering: To ensure statistical validity, stocks with excessive missing information (more than 3 null metrics) were excluded from the analysis. This rigorous filtering process reduced the dataset from the initial 194 tickers to 169 qualified issuers, ensuring that only stocks with sufficient fundamental data were processed.

2) Directionality Adjustment: SVD assumes larger scalar values indicate stronger "features." However, for valuation metrics (e.g., PE Ratio), a lower value is better. Therefore, we invert the logic for "Lower-is-Better" metrics by multiplying by -1.
3) Z-Score Standardization: Financial metrics have vastly different units (e.g., Market Cap in trillions vs. ROE in decimals). To prevent metrics with large nominal values from dominating the variance, all columns are standardized to have zero mean and unit variance:

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

D. Singular Value Decomposition (SVD)

Singular Value Decomposition is a fundamental theorem which states that any real matrix A of size $m \times n$ can be factored into three matrices [4], [5]:

$$A = U \Sigma V^T \tag{3}$$

Where:

- U $(m \times m)$: Orthogonal matrix representing the relationship between Stocks (rows) and latent concepts. In this financial context, the rows of U indicate how strongly each stock aligns with the identified market factors (e.g., Quality, Value).
- Σ $(m \times n)$: Diagonal matrix containing singular values σ_i sorted in descending order ($\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$). These values quantify the "strength" or variance explained by each latent concept.
- V^T $(n \times n)$: Orthogonal matrix representing the relationship between Metrics (columns) and latent concepts. The values in V^T reveal the weight or contribution of each financial ratio (e.g., how much ROE contributes) to the overall quality score.

E. Outer Product Expansion and Ranking

The matrix A can be expressed as the sum of rank-1 matrices derived from the outer product of the left and right singular vectors [4]:

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \dots \tag{4}$$

Here, the term $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ captures the largest variance (the dominant trend) of the dataset. For stock ranking, we extract the first column of U ($\mathbf{u}_1$). This vector serves as a "Latent Quality Score," effectively reducing the 12-dimensional complexity into a single scalar value for ranking.

III. IMPLEMENTATION PROGRAM

The automated stock ranking system is implemented using the Python programming language, utilizing a modular approach to ensure scalability. The system relies on four primary libraries: yfinance for retrieving market data [7], pandas for structured data manipulation, numpy for high-performance linear algebra computations, and seaborn for statistical data visualization. The execution flow is divided into four distinct stages: data acquisition, preprocessing, matrix decomposition, and result visualization.

A. Data Acquisition and Local Caching

The program begins by defining a target universe of stocks. A list of 194 tickers, representing a broad cross-section of the Indonesia Stock Exchange (IDX) including the LQ45 and Kompas100 indices, is hardcoded into the system.

To mitigate network latency and API rate limits, a caching mechanism is implemented. The program checks for the existence of a local storage file (stockData.csv).

- Cache Hit: If the file exists, data is loaded immediately into a DataFrame.
- Cache Miss: If the file is absent, the program initiates a loop using tqdm to iterate through each ticker, fetching 12 specific financial keys via the Yahoo Finance API. The results are then serialized to CSV for future use.

Fig. 3.1: Implementation of the data fetching loop and CSV caching logic.

Source: Author

B. Data Preprocessing and Vector Space Construction

Raw financial data is inherently noisy and heterogeneous. The preprocessing module transforms this raw data into a mathematically rigorous matrix A through three distinct steps:

1) Filtering and Imputation: To ensure data integrity, the system first performs a validity check. Stocks with excessive missing values (missing more than 3 metrics) are discarded from the dataset. For the remaining stocks, statistical imputation is applied by filling null values with the *median* of their respective columns. Median imputation is selected over the mean to minimize the skewing impact of extreme outliers often found in financial data.
2) Directionality Adjustment: SVD treats magnitude as "strength." However, metrics like PE Ratio, Price-to-Book (PBV), and Debt-to-Equity (DER) operate on a

"lower-is-better" basis. To align these with "higher-is-better" metrics (like ROE), the program multiplies the columns of valuation and risk metrics by -1. This ensures that in the final vector space, a higher scalar value universally represents a more positive attribute.

3) Z-Score Standardization: The data is standardized to have a mean of 0 and a standard deviation of 1. This prevents metrics with large nominal values (such as Market Cap in trillions of Rupiah) from dominating the variance structure of the SVD calculation.

```python
df_clean = df_raw.dropna(thresh=(len(metrics_config) - 3))
df_imputed = df_clean.fillna(df_clean.median())
df_adjusted = df_imputed.copy()

for col in lower_is_better:
    df_adjusted[col] = df_adjusted[col] * -1

df_standardized = (df_adjusted - df_adjusted.mean()) / df_adjusted.std()
```

Fig. 3.2: Code handling data filtering, directionality inversion, and Z-score standardization.

Source: Author

C. SVD Computation and Feature Extraction

The normalized data is converted into a Numpy matrix A. The program then executes the Singular Value Decomposition:

$$U, S, V^T = \text{numpy.linalg.svd}(A) \tag{5}$$

This stage performs two critical analytical functions:

1) Stock Ranking (U Matrix): The first column of matrix U is extracted as the latent quality score. To ensure the correct orientation (i.e., verifying that positive scores indicate good fundamentals), the program calculates the correlation between the SVD score and Return on Equity (ROE). If the correlation is negative, the score vector is multiplied by -1.
2) Feature Importance (V^T Matrix): The program analyzes the first row of the V^T matrix to determine the "Feature Weights." This allows the system to quantify which financial metrics (e.g., Liquidity vs. Profitability) contribute most significantly to the ranking, providing interpretability to the mathematical model.

```python
A = df_standardized.values
U, S, Vt = np.linalg.svd(A, full_matrices=False)
quality_scores = U[:, 0]

if np.corrcoef(quality_scores, df_standardized['ROE'])[0, 1] < 0:
    quality_scores *= -1
    Vt[0] *= -1
```

Fig. 3.3: SVD execution, sign alignment logic, and feature weight extraction.

Source: Author

D. Visualization and Result Interpretation

The final module presents the analysis results in two formats. First, it extracts and prints the Top 5 "Quality Stocks" and Bottom 5 "High-Risk Stocks" based on the computed SVD scores.

Second, the system generates graphical visualizations to aid interpretation:

- Ranking Bar Chart: A color-coded bar chart (green for positive, red for negative) visualizes the distribution of scores for the top and bottom outliers.
- Correlation Heatmap: A heatmap of the 12×12 correlation matrix is generated to visualize the redundancy between metrics (multicollinearity), thereby justifying the use of SVD for dimensionality reduction.

```python
df_final = df_clean.copy()
df_final['SVD_Score'] = quality_scores
df_final['Rank'] = df_final['SVD_Score'].rank(ascending=False).astype(int)
df_final = df_final.sort_values('Rank')

# Hitung bobot kontribusi setiap metrik terhadap skor
feature_weights = pd.Series(Vt[0], index=df_standardized.columns).sort_values(ascending=False)

# 6. RESULT
def print_header(title):
    print(f"\n{'='*80}\n {title.upper()} \n{'='*80}")

print_header(f"ANALYSIS COMPLETE: {len(df_final)} STOCKS RANKED")

print("\nMETRIC IMPORTANCE (WEIGHTS IN SVD VECTOR)")
print("(Positive = Drives Rank Up | Negative = Drives Rank Down)")
print("-" * 60)
print(feature_weights.to_string(float_format="{:,.4f}".format))

cols_output = ['Rank', 'SVD_Score', 'ROE', 'PE_Ratio', 'DER', 'PBV']

print_header("TOP 5 QUALITY STOCKS (UNDERVALUED & PROFITABLE)")
print(df_final[cols_output].head(5).to_string(index=True, float_format="{:,.2f}".format))

print_header("BOTTOM 5 STOCKS (RISKY / OVERVALUED)")
print(df_final[cols_output].tail(5).to_string(index=True, float_format="{:,.2f}".format))


# 7. VISUALIZATION
plt.style.use('dark_background')
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 8))

# A. Barplot Top & Bottom 5
viz_data = pd.concat([df_final.head(5), df_final.tail(5)])
colors = ['#00ff00' if x >= 0 else '#ff3333' for x in viz_data['SVD_Score']] # Neon Green & Red

sns.barplot(x=viz_data['SVD_Score'], y=viz_data.index, palette=colors, ax=ax1, edgecolor='white')
ax1.set_title('★ Top 5 vs Bottom 5 Stocks by SVD Score', fontsize=16, fontweight='bold', color='white')
ax1.set_xlabel('Latent Quality Score (SVD)', fontsize=12)
ax1.axvline(0, color='white', linewidth=1, linestyle='--')
ax1.grid(axis='x', alpha=0.1)
```

Fig. 3.4: Code for sorting data and plotting the ranking results.

Source: Author

```python
corr = df_standardized.corr()
mask = np.triu(np.ones_like(corr, dtype=bool))

sns.heatmap(corr, mask=mask, cmap='RdYlGn', center=0, square=True, linewidths=.5, cbar_kws={"shrink":
.5}, ax=ax2)
ax2.set_title('Correlation Matrix of Financial Metrics', fontsize=16, fontweight='bold', color='white')

plt.tight_layout()
plt.show()
```

Fig. 3.5: Code for correlation matrix heatmap.

Source: Author

IV. ANALYSIS

This section interprets the computational results generated by the SVD algorithm. It analyzes the mathematical significance of the feature weights, the fundamental characteristics of the ranked stocks, and the structural implications of the data distribution.

A. Latent Factor Interpretation

The primary advantage of using Singular Value Decomposition (SVD) is its ability to reveal the underlying structure of the data through the right singular vectors (V^T). Instead of treating all metrics equally, the model objectively assigns weights based on the maximum variance found in the normalized dataset.

As shown in the program output (Fig. 4.1), the first principal component, which serves as the "Latent Quality Score", is driven by three distinct factors:

- Liquidity Dominance: Current Ratio (0.4392) and Quick Ratio (0.4358) possess the highest positive coefficients. Mathematically, this indicates that the variation in cash positions constitutes the largest source of information in the dataset. Consequently, the model implicitly defines "Quality" as the ability to maintain short-term solvency.
- Efficiency Drivers: Gross Margin (0.4201) and Operating Margin (0.4051) also contribute significantly, filtering out low-margin businesses regardless of their revenue size.
- Valuation Sensitivity: The positive weight on the inverted PBV metric implies that the model strictly rewards lower valuations.

```
METRIC IMPORTANCE (WEIGHTS IN SVD VECTOR)
(Positive = Drives Rank Up | Negative = Drives Rank Down)
-----------------------------------------------------------
Current_Ratio   0.4392
Quick_Ratio     0.4358
Gross_Margin    0.4201
Opr_Margin      0.4051
ROA             0.3189
PBV             0.2398
EV_EBITDA       0.1680
ROE             0.1643
FCF             0.1380
PS_Ratio        0.1372
DER             0.1172
PE_Ratio        0.1026
```

Fig. 4.1: Console output showing the feature weights.

Source: Author

B. Analysis of Stock Rankings

The model generated a ranking of 169 stocks based on the projection of each stock vector onto the first principal component. Fig. 4.2 displays the outliers identified by the system.

1) Top 5 Quality Stocks: The top 5 stocks (CFIN, BUKA, SIDO, NICL, ADMF) reached the summit through different mathematical paths. They can be categorized into three profiles:

- Deep Value Plays (CFIN, ADMF): These multi-finance companies topped the list primarily due to the Valuation factor. They trade at very low Price-to-Book Values (PBV < 1.0). In the vector space, the inverted PBV metric creates a massive positive magnitude, propelling them to the top.
- Liquidity-Driven Selection (BUKA): Bukalapak (Rank 2) is a notable inclusion driven by the model's heavy bias towards liquidity. Despite mixed profitability, BUKA holds massive cash reserves. Since the model assigned a 0.43 weight to the Current Ratio, BUKA's exceptional balance sheet strength mathematically "overpowered" its lower earnings scores.
- The High-Performance Anomaly (NICL, SIDO): SIDO (Rank 3) and NICL (Rank 4) represent the "Quality" segment. Interestingly, NICL trades at a high valuation (PBV $> 14x$). However, it remains in the Top 5 because its Return on Equity (ROE) was exceptionally high (approx. 66%). The magnitude of its Profitability vector was strong enough to offset the penalty from its high Valuation vector.

2) Bottom 5 Risky Stocks: The bottom 5 stocks (BRPT, DOID, MPPA, KOBX, TPIA) illustrate the model's penalty mechanism.

- The Valuation Trap: Stocks like TPIA and BRPT possess extreme PBV ratios (often exceeding 100x). The inverted scoring logic transformed these values into large negative scores, sending them to the bottom of the list.
- Financial Distress: Companies like MPPA and DOID were penalized due to a "double drag": negative earnings (reducing the ROE component) combined with high leverage (increasing the DER penalty).

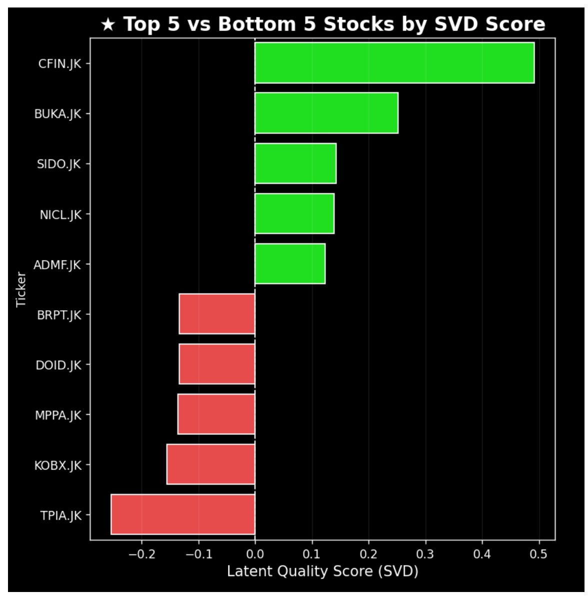


Fig. 4.2: Program execution result displaying the Top 5 and Bottom 5 ranked stocks.

Source: Author

C. Visualization of the Valuation Trade-off

Blue Chip stocks, particularly major banks like BBCA and BBRI, are traditionally regarded as the "Gold Standard" of the Indonesia Stock Exchange. Characterized by consistent earnings, market dominance, and high liquidity, they typically command the highest investor confidence. Intuitively, one might expect these market leaders to dominate any fundamental ranking system.

However, the SVD model reveals a critical distinction between "Business Quality" and "Investment Value." As visualized in the scatter plot (Fig. 4.3), the results show a clear separation based on valuation premiums:

- The Blue Chip Paradox (Blue Dots): Major banks are clustered in the middle-right section of the plot. While they are fundamentally strong, they trade at premium valuations (High PBV). The model implicitly suggests that while these companies are excellent businesses, they are currently expensive investments.
- The Value Winners (Green Dots): The Top 5 stocks are primarily clustered on the left side (Low PBV). This confirms that the model is strictly biased towards "Bargain Hunting", identifying companies that are fundamentally sound but priced cheaply by the market.
- The Profitability Outlier (NICL): The notable exception is NICL (Green dot on the far right). Despite having a valuation even higher than Blue Chips, it secured a Top 5 spot. This proves that a stock *can* overcome the valuation penalty if its profitability metrics (ROE) are exceptionally high enough to compensate for the premium price.

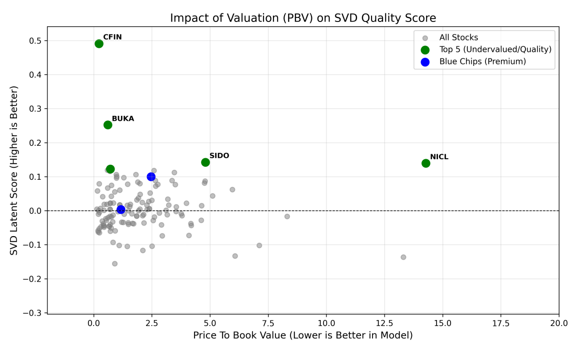


Fig. 4.3: Scatter plot of PBV vs. SVD Score.

Source: Author

D. Multicollinearity and Heatmap Analysis

The necessity of using SVD is further justified by the correlation matrix of the financial metrics. As visualized in Fig. 4.4, the dataset exhibits distinct "clusters" of information redundancy that would impair traditional linear models.

Two primary clusters of multicollinearity are evident. First is the Liquidity Cluster, where Current Ratio and Quick Ratio exhibit a correlation coefficient nearing 1.0 (Dark Red). This indicates that for most companies, these two metrics provide nearly identical information regarding short-term solvency. Second is the Efficiency Cluster, shown by the strong positive correlation between Gross Margin and Operating Margin. Without dimensionality reduction, these correlated variables would artificially inflate the importance of liquidity and margins, leading to a biased ranking system ("double counting" effect).

SVD resolves this issue through orthogonal transformation. By decomposing the matrix A into $U\Sigma V^T$, the algorithm constructs principal components that are mathematically orthogonal (uncorrelated) to each other. This process effectively compresses the redundant dimensions, such as the liquidity pair, into a single, robust latent factor. Consequently, the final "Quality Score" represents a clean synthesis of the data's variance, free from the noise and instability caused by multicollinearity.

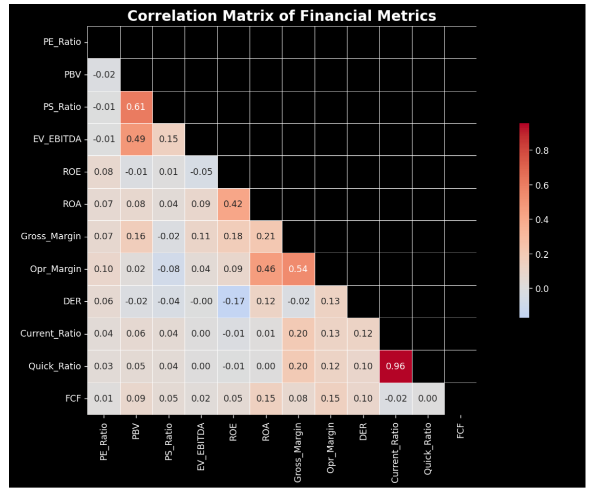


Fig. 4.4: Correlation matrix heatmap.

Source: Author

V. CONCLUSION

This study demonstrates that Singular Value Decomposition (SVD) serves as a powerful, objective framework for automating stock selection. By transforming 194 stocks into vectors within a 12-dimensional Euclidean space, the model successfully synthesized complex financial data into a single "Latent Quality Score." The algorithm naturally prioritized Liquidity and Efficiency as the primary drivers of corporate quality, effectively identifying "Hidden Gems" like CFIN and NICL, stocks that offer a rare combination of deep undervaluation and high profitability, while strictly penalizing premium valuations found in traditional Blue Chips.

While the model eliminates human emotional bias, it exhibits a distinct "Value Bias," making it most suitable for contrarian investing strategies rather than momentum trading. Future enhancements should consider sector-specific normalization to address the liquidity metrics bias against banking institutions. Ultimately, this research confirms that linear algebra techniques provide a rigorous mathematical foundation for modern financial technology, enabling investors to navigate the curse of dimensionality with computational precision.

VI. APPENDIX

The program source code used in this paper is available at GitHub Repository, and the video demonstration can be viewed at Youtube.

ACKNOWLEDGMENT

First and foremost, I would like to thank Jesus for his blessings and grace, which enabled me to complete this paper. I would also like to express my sincere gratitude to Ir. Rila Mandala, M.Eng., Ph.D. and Dr. Ir. Rinaldi Munir, M.T. as the lecturers of IF2123 Linear Algebra and Geometry for their guidance and the valuable knowledge shared during the Informatics Engineering class. Additionally, the author acknowledges the use of computational assistants and AI-based tools for code debugging and linguistic refinement during the preparation of this manuscript.

REFERENCES

[1] R. Munir, "IF2123 Aljabar Linier dan Geometri," Sekolah Teknik Elektro dan Informatika ITB, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/algeo25-26.htm. [Accessed: Dec. 20, 2025].

[2] R. Munir, "Vektor di ruang Euclidean (Bagian 1-3)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2025.

[3] R. Munir, "Ruang vektor umum (Bagian 1-4)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2025.

[4] R. Munir, "Singular Value Decomposition (SVD) (Bagian 1-2)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2025.

[5] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley, MA: Wellesley-Cambridge Press, 2016.

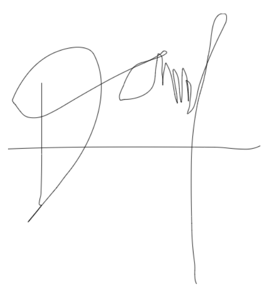
[6] B. Graham and D. L. Dodd, *Security Analysis*, 6th ed. New York: McGraw-Hill, 2008.

[7] R. Aroussi, "yfinance: Yahoo! Finance market data downloader," PyPI, 2023. [Online]. Available: https://pypi.org/project/yfinance/.

DECLARATION OF ORIGINALITY

I hereby declare that this paper is my own original work, not an adaptation or translation of another person's paper, and is free from plagiarism.

Bandung, December 20, 2025



Daniel Anindito Nugroho
13524002