Implementation of a System for Clustering and Stability Analysis of Price Dynamics of Traditional Market Commodities Based on Vector Spaces and Eigenvalues

Niko Samuel Simanjuntak - 13524029
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13524029@std.stei.itb.ac.id, nikosimanjuntak71@gmail.com

***Abstract*—Commodity price dynamics in traditional markets are often highly fluctuative and difficult to map manually, creating uncertainty for traders and market administrators in mitigating inflationary risk. This paper implements a commodity clustering and price stability analysis system using a linear algebra–based approach grounded in vector spaces and eigenvalues. By transforming historical price data into logarithmic returns, each commodity is represented as a statistical feature vector in a five-dimensional Euclidean space ($\mathbb{R}^5$) encompassing parameters such as mean, volatility, and trend. This representation enables automatic clustering using the K-Means algorithm to objectively distinguish between stable and volatile commodities. Furthermore, the aggregate price dynamics of each cluster are modeled as a discrete linear system, in which system stability is evaluated based on the spectral radius of the corresponding transition matrix. Experimental results on a simulated dataset demonstrate that the proposed approach is capable of accurately identifying market behavior structures and confirm that eigenvalues can serve as precise quantitative indicators for detecting potential instability or price divergence within specific commodity groups.**

***Keywords*—Vector Space, Eigenvalues, Dynamical System Stability, Logarithmic Return, Traditional Markets**

I. INTRODUCTION

Traditional markets play a vital role in Indonesia's food distribution chain and grassroots economy. Unlike modern retail markets, which operate with centralized inventory management and relatively stable prices, commodity price dynamics in traditional markets tend to be highly fluctuative and stochastic. These price changes are driven by complex interacting factors, ranging from seasonal supply availability and distribution costs to shifting local demand trends.

In practice, traders and market administrators often face difficulties in identifying patterns of relationships among commodities. Price movements in one commodity (for example, red chili) may exhibit similar risk patterns to those of another commodity (such as shallots), yet differ substantially from relatively stable staple commodities (such as rice). In the absence of adequate analytical tools, such commodity groupings are frequently performed based on intuition alone, making it difficult to obtain a measurable understanding of which commodity groups tend to move together (are correlated) and which exhibit relatively independent dynamics. Furthermore, simple yet systematic algorithmic approaches for early detection of price dynamic "stability" are still rarely applied in the context of traditional markets. In fact, information regarding whether a price shock is temporary or has the potential to develop into persistent inflation is crucial for operational decision-making.

This paper proposes a linear algebra–based approach that leverages the concepts of vector spaces and eigenvalues to address these challenges. In the first stage, historical price data are transformed into logarithmic returns to normalize scale differences. Each commodity is then represented as a statistical feature vector in a high-dimensional Euclidean space ($\mathbb{R}^n$). The axes of this space correspond to price behavior parameters such as average growth, volatility, and linear trend, allowing similarities among commodities to be quantified using Euclidean distance. This representation enables a clustering process that groups commodities with similar fluctuation patterns in a more objective manner than intuitive classification.

In the second stage, the temporal price dynamics within each group are modeled as a discrete linear dynamical system through a transformation matrix. System stability is then analyzed using the spectral properties of this matrix. In this study, stability is operationally defined based on the spectral radius (the maximum absolute value of the eigenvalues): a commodity group is considered stable if the magnitudes of all eigenvalues of its dynamical matrix are less than one ($|\lambda| < 1$), indicating that price disturbances tend to decay over time.

The main objective of this paper is to implement a simple system capable of clustering traditional market commodities based on their statistical price patterns and analyzing the stability of price dynamics at the group level. The system is constructed as a computational pipeline encompassing data preprocessing, feature vector extraction, clustering, dynamic matrix estimation, and eigenvalue computation. The primary contribution of this work lies in the application of fundamental linear algebra theory to simple microeconomic modeling, as well as in demonstrating that abstract concepts such as eigenvectors can be translated into practical indicators for monitoring the health of commodity price dynamics. This approach is expected to provide measurable analytical insights into market structure and serve as a foundation for the development of intelligent price monitoring systems in the future.

II. Literature Review

A. Vectors and Feature Space

Geometrically, a vector in the Euclidean space $\mathbb{R}^n$ is defined as a directed line segment that has both a length (magnitude) and a direction. However, in the context of computer science and data analysis, a vector is viewed algebraically as an n-tuple, namely an ordered collection of n real numbers arranged as a single entity:

$$\boldsymbol{v} = [v_1, v_2, \dots, v_n]^T$$

In this study, a market commodity (for example, *Red Chili* or *IR64 Rice*) is not represented as a single scalar value, but rather as a feature vector $\mathbf{x} \in \mathbb{R}^d$. The components of this vector correspond to statistical parameters extracted from the historical price behavior of the commodity. If we employ $d = 3$ main features, namely the average price (μ), volatility (σ), and upward trend (m), then each commodity can be mapped to a single point in a three-dimensional feature space:

$$\boldsymbol{x}_{\text{commodities}} = (\mu, \sigma, m)$$

This mapping allows the use of the Euclidean distance concept to quantify the "similarity" between commodities. Two commodities that have a small Euclidean distance in the feature space are considered to exhibit similar market characteristics, even if the types of goods themselves are different.

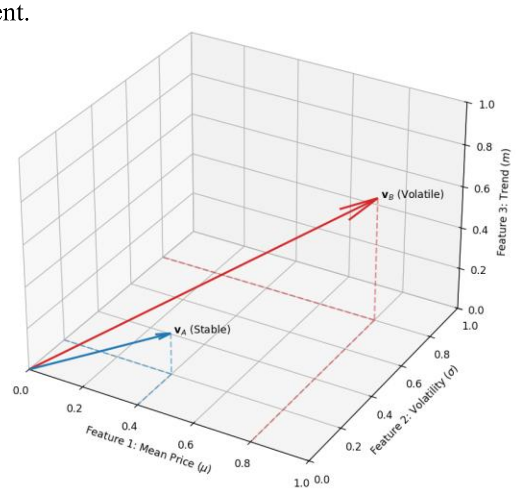

Figure 1. Representation of Commodities as Vectors in Three-Dimensional Feature Space [Source: personal documentation]

B. Logarithmic Return

Price data of commodities in traditional markets often exhibit highly heterogeneous value ranges. For example, the price of beef (≈ IDR 140,000/kg) is significantly higher than the price of spinach (≈ IDR 5,000 per bunch). If analysis is performed directly on nominal prices, high-priced commodities tend to dominate variance calculations and distort the results of clustering.

To address this issue, a logarithmic return transformation is employed. Let P_t denote the price at day t and P_{t-1} the price on the previous day. The log-return r_t is defined as:

$$r_t = \ln\left(\frac{P_t}{P_{t\text{-}1}}\right) = \ln(P_t) - \ln(P_{t\text{-}1})$$

This transformation serves two crucial functions within the system:

1. Scale Normalization: It converts absolute price values into relative percentage changes, allowing price fluctuations of commodities such as spinach and beef to be compared fairly on a like-for-like basis.
2. Statistical Properties: The distribution of log-returns tends to be more symmetric and closer to a normal distribution, which is an important assumption in many linear and statistical modeling approaches.

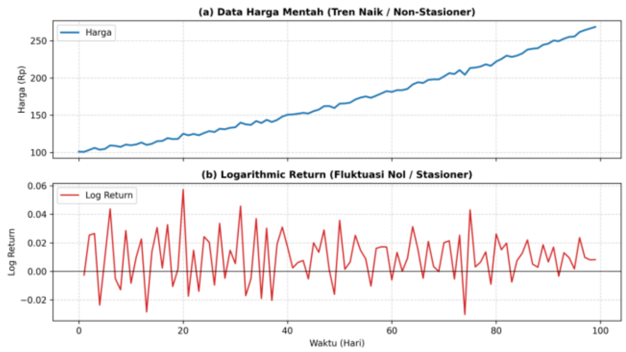

Figure 2. Comparison of Data Transformations [Source: personal documentation]

C. Matrices and Linear Transformations

A matrix is an arrangement of scalar values in a rectangular form, organized into rows and columns. A square matrix $A \in \mathbb{R}^{n\times n}$ plays a central role in linear algebra as a linear transformation operator. Such a matrix maps an input vector x to an output vector y through matrix multiplication:

$$y = Ax$$

In the context of market price dynamics, matrices are used to model interactions and transitions. If s_t denotes the market state vector at the current day (containing a statistical summary of all commodities), then the matrix A represents the market mechanism that transforms today's state into tomorrow's state (s_{t+1}). The elements of matrix A implicitly encode information about correlations and cross-influences among different groups of commodities.

D. Eigenvalues and Eigenvectors

The concept of eigenvalues lies at the core of stability analysis in this study. Let A be a square matrix of size $n \times n$. A scalar λ is called an eigenvalue, and a non-zero vector v is called an eigenvector, if they satisfy the fundamental equation:

$$A\mathrm{v} = \lambda \mathrm{v}$$

Intuitively, this equation states that an eigenvector v represents a special direction in which the linear transformation A acts only as a scaling operation (either stretching or shrinking) without changing the direction of the vector. The corresponding scaling factor is the eigenvalue λ.

Eigenvalues are obtained by finding the roots of the characteristic polynomial:

$$\det(A - \lambda I) = 0$$

In the context of price data analysis:

- Eigenvectors represent the "modes" or dominant patterns of market price movements (for example, a direction in which the prices of all commodities increase simultaneously).
- Eigenvalues indicate the strength or dominance of these patterns and determine whether the corresponding modes will grow or decay over time.

E. Principal Component Analysis (PCA)

Because the commodity feature vectors in this study have dimensionality $d > 2$ (i.e., they are multidimensional), it is difficult for humans to directly observe clustering patterns. Principal Component Analysis (PCA) is therefore employed as a dimensionality reduction technique for visualization purposes.

PCA works by finding a new set of basis vectors that maximize the variance of the data. These basis vectors are, in fact, the eigenvectors of the data covariance matrix.

The PCA procedure can be summarized as follows:

1. The covariance matrix C is computed from the normalized feature data.
2. Eigen decomposition is performed on C to obtain pairs $(\lambda_i, \mathrm{v}_i)$.
3. The eigenvectors corresponding to the largest eigenvalues (v_1 and v_2) are selected as the main axes, known as the principal components.

Projecting the data onto the subspace spanned by v_1 and v_2 ensures that the original clustering structure (such as clusters of chili prices versus rice prices) is preserved as much as possible in a two-dimensional representation.

F. Stability Analysis of Discrete Dynamical Systems

One of the main objectives of this paper is to determine whether market price fluctuations tend to be stable or to exhibit explosive (volatile) behavior. This phenomenon is modeled using a linear discrete dynamical system:

$$\mathrm{s}_{t+1} = A\mathrm{s}_t$$

where s_t denotes the state vector of commodity clusters at time t. The stability of this system is entirely determined by the eigenvalue spectrum of the transition matrix A. The key parameter is the spectral radius (ρ), which is defined as the largest absolute value among all eigenvalues:

$$\rho(A) = \max\{|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|\}$$

The economic interpretation of the spectral radius is as follows:

- If $\rho(A) < 1$ (Asymptotically Stable System): Price disturbances (shocks) will gradually dissipate over time. Prices will converge back to an equilibrium point (long-term average), indicating a healthy and stable market.
- If $\rho(A) > 1$ (Unstable System): Price disturbances will be amplified (grow exponentially) over time. This indicates the presence of uncontrolled inflation or highly explosive price volatility (a market bubble).

G. K-Means Clustering

K-Means is an iterative clustering algorithm that partitions a dataset into K distinct clusters $(C_1, C_2, \dots, C_K)$. The objective of this algorithm is to minimize the within-cluster variance (intra-cluster variance), which is formulated as the sum of squared Euclidean distances between each data point xand its corresponding cluster center (centroid μ_k):

$$J = \sum_{k=1}^{K} \sum_{\mathrm{x} \in C_k} \| \mathrm{x} - \mu_k \|^2$$

In this paper, *K-Means* is applied to commodity feature vectors. The result is an automatic clustering process in which commodities with similar statistical behavior (for example, a group of "volatile kitchen spices" versus "stable staple goods") are placed into the same cluster without any prior manual labeling.

III. Implementation

The system implementation is developed using the Python programming language. This choice is motivated by Python's strong scientific computing ecosystem, which enables a direct translation of matrix and vector notation into computational data structures. The main libraries used include NumPy for high-performance linear algebra operations, Pandas for time-series data manipulation, Matplotlib for graphical visualization, and modules from Scikit-learn for eigenvalue-based decomposition (PCA) and clustering algorithms.

Overall, the software architecture consists of four sequential processing modules:

(A) Data Preprocessing and Logarithmic Transformation,

(B) Feature Vector Space Construction,

(C) Commodity Clustering Based on Euclidean Distance, and

(D) Stability Analysis of the Dynamical System.

A. Data Preprocessing: Log-Return Transformation

The raw data obtained from traditional markets consist of daily transaction records containing nominal prices denominated in Indonesian Rupiah. This data exhibits two characteristics that complicate direct analysis:
(1) a very wide range of scales (price heterogeneity), and
(2) non-stationarity, where prices tend to exhibit a long-term upward trend due to inflation.

To address these issues, the first step in the system is to transform the nominal price series into a logarithmic return series (r_t). This transformation normalizes the data so that the measured quantity is no longer "how much the price is," but rather "how fast the price changes."

```
def compute_log_returns(price_series:
pd.Series) -> pd.Series:
    """
    Computes the log-return: r_t = ln(p_t) -
ln(p_{t-1}).
    This method normalizes the price scale so
that expensive and inexpensive
    commodities can be compared fairly.
    """
    p = price_series.astype(float)

    # [1] Handling Data Discontinuities
    # Replace zero values with NaN and apply
forward fill to maintain continuity.
    # This prevents the mathematical error
log(0), which is undefined.
    p = p.replace(0,
np.nan).ffill().fillna(1.0)

    # [2] Logarithmic Differencing
    # np.log(p) computes ln(p_t), and .diff()
computes the difference with ln(p_{t-1})
    return np.log(p).diff().fillna(0.0)
```

In the code block above, line [1] serves as a defensive programming mechanism. Market data are often imperfect (for example, when a trader closes for a day), which may result in recorded prices of zero. This code ensures that missing or invalid values are replaced with the last known price (*forward fill*), thereby preserving the assumption of market continuity.

Line [2] is the direct implementation of the formula
$r_t = \ln(P_t) - \ln(P_{t-1})$.
This function transforms a price series that may exhibit an exponential trend into a fluctuation series centered around zero, which is an ideal prerequisite for subsequent statistical analysis.

B. Feature Space Construction

This stage constitutes the core implementation of the "feature space–based" concept. If raw price data over 100 days were used directly as a vector, the analysis would take place in a 100-dimensional space ($\mathbb{R}^{100}$), which is highly susceptible to the curse of dimensionality and daily noise.

Instead, the system maps each commodity into a fixed, low-dimensional feature space ($\mathbb{R}^5$). Each feature vector x_i is constructed by extracting five key statistical parameters from the log-return series of each commodity. These parameters are selected for their ability to comprehensively represent the market "behavior" of a commodity.

```
def extract_features_from_series(r: pd.Series)
-> dict:
    """
    Constructs a commodity feature vector x_i
∈ R^d from the return series r(t).
    These features represent the 'behavior' of
commodity prices in the vector space.
    """
    y = r.to_numpy(dtype=float)
    T = len(y)

    # [1] First and Second Moment Statistics
(Mean & Volatility)
    # Mean reflects directional bias
(upward/downward tendency)
    # Std reflects price risk or uncertainty
    mean = float(np.mean(y))
    std = float(np.std(y, ddof=1)) if T > 1
else 0.0
    var = std ** 2

    # [2] Linear Trend Analysis (Slope)
    # Uses covariance to estimate the slope of
return regression over time
    t_arr = np.arange(T, dtype=float)
    if T >= 2:
        cov_ty = float(np.cov(t_arr, y,
ddof=1)[0, 1])
        var_t = float(np.var(t_arr, ddof=1))
        slope = cov_ty / var_t if var_t > 0
else 0.0
    else:
        slope = 0.0

    # [3] Lag-1 Autocorrelation (System
Memory)
    # Measures how strongly today's price is
influenced by yesterday's price
    if T >= 3 and np.std(y[:-1]) > 1e-12 and
np.std(y[1:]) > 1e-12:
        acf1 = float(np.corrcoef(y[:-1],
y[1:])[0, 1])
    else:
        acf1 = 0.0

    return {
        "mean": mean,
        "std": std,
        "var": var,
        "slope": slope,
        "acf1": acf1
    }
```

This function deterministically transforms a time-series of returns into a feature vector

$$\mathrm{x} = [\mu, \sigma, \sigma^2, m, \rho_{\text{lag1}}].$$

- Mean (μ) and Slope (m): Capture long-term trends. Commodities with high inflationary behavior tend to exhibit large positive values along these dimensions.
- Standard Deviation (σ) and Variance (σ^2): Capture risk-related characteristics. In the feature space, these metrics help distinguish stable commodities (such as rice) from volatile ones (such as chili).
- Lag-1 Autocorrelation (ρ_{lag1}): Captures temporal dependence patterns. Values close to 1 indicate strong price inertia (tomorrow's price is very similar to today's), whereas values close to 0 indicate random-walk behavior.

C. Commodity Clustering

After each commodity is represented as a point in the $\mathbb{R}^5$ feature space, the next step is to group these points into clusters. However, because each feature has a different scale (for example, variance values are much smaller than autocorrelation values), a standardization process (Z-score normalization) is applied first.

The K-Means algorithm is then used to partition the vector space. To avoid the subjective selection of the number of clusters (k), the system implements an automatic search for the optimal value of k by maximizing the Silhouette Score.

```
def choose_k_by_silhouette(X: np.ndarray,
k_min: int = 2, k_max: int = 8, seed: int =
2123) -> int:
    """
    Selects the optimal number of clusters
(k) using the Silhouette Score.
    The Silhouette Score measures how well an
object fits within its own cluster (cohesion)
    compared to neighboring clusters
(separation).
    """
    best_k = k_min
    best_score = -1.0

    # Limit k_max so that it does not exceed
the number of available data samples
    max_possible = min(k_max, X.shape[0] - 1)
    if max_possible < k_min:
        return k_min

    for k in range(k_min, max_possible + 1):
        # [1] K-Means Initialization and
Model Fitting
        km = KMeans(n_clusters=k,
random_state=seed, n_init=10)
        labels = km.fit_predict(X)

        # Stop if the clustering result is
trivial (only one cluster is formed)
        if len(set(labels)) < 2:
            continue

        # [2] Evaluation of Geometric
Partition Quality
        score = silhouette_score(X, labels)
        if score > best_score:
            best_score = score
            best_k = k

    return best_k
```

This function performs an iterative trial over different values of k. At each iteration, the K-Means algorithm attempts to minimize inertia (intra-cluster distance). However, inertia alone is insufficient, because it always decreases as $k$increases. Therefore, the Silhouette Score is employed, as it jointly captures cluster cohesion and separation. The value of k that yields the highest silhouette score is considered the most natural geometric representation of the underlying data structure.

D. Stability Analysis of the Dynamical System

This section implements a discrete dynamical system model for each cluster that is formed. It is assumed that the aggregate price dynamics within a cluster follow a linear relationship

$$s_{t+1} = As_t,$$

where $s_t$denotes the cluster state vector at time t.

```
def estimate_var1_matrix(S: np.ndarray,
ridge_eps: float = 1e-6) -> np.ndarray:
    """
    Estimates the transition matrix A using
Least Squares with Ridge Regularization:
    Minimizes || Y - A Z ||^2 -> Solution: A
= Y Z^T (Z Z^T + epsilon I)^-1
    """
    T, m = S.shape
    if T < 3:
        return np.eye(m)  # Fallback if
historical data are insufficient

    Z = S[:-1].T  # State matrix at time t
(input)
    Y = S[1:].T   # State matrix at time t+1
(target)

    # [1] Auto-covariance Matrix Computation
    ZZt = Z @ Z.T

    # [2] Matrix Inversion with
Regularization
    # Adding ridge_eps to the main diagonal
guarantees invertibility (non-singular
matrix)
    A = (Y @ Z.T) @ np.linalg.inv(ZZt +
ridge_eps * np.eye(m))
    return A

def spectral_radius_and_eigs(A: np.ndarray) -
> tuple[float, np.ndarray]:
    """
    Computes the spectral radius (maximum
absolute eigenvalue).
    rho < 1 : Stable system (disturbances
decay over time)
    rho > 1 : Unstable system (disturbances
grow over time)
    """
    # [3] Eigenvalue Decomposition
    eigvals = np.linalg.eigvals(A)

    # [4] Maximum Stability Determination
    rho = float(np.max(np.abs(eigvals)))
    return rho, eigvals
```

This code is a direct translation of least squares theory.

- In lines [1] and [2], the system estimates the transition matrix A that best predicts tomorrow's state based on today's state. The use of Ridge regularization (ridge_eps) is a crucial numerical technique to stabilize the matrix inversion when the data exhibit high multicollinearity.
- In line [3], the function np.linalg.eigvals computes the roots of the characteristic polynomial of matrix A, i.e., its eigenvalues.
- Line [4] computes the spectral radius (ρ), which serves as a single indicator of system stability.

E. Result Visualization: PCA Projection

Because the feature vector space has dimensionality $d = 5$, direct visualization on a computer screen is not feasible. For reporting and visual analysis purposes, Principal Component Analysis (PCA) is employed to project the data onto a two-dimensional plane ($d = 2$).

```
def visualize_results(X: np.ndarray, labels: np.ndarray, ...):
    # [1] Dimensionality Reduction
    # Project data from R^5 to R^2 using the two dominant eigenvectors
    pca = PCA(n_components=2)
    X_pca = pca.fit_transform(X)

    # [2] Scatter Plotting
    plt.figure(figsize=(14, 6))
    plt.subplot(1, 2, 1)
    scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=labels, cmap='viridis', ...)

    # Add axis labels with explained variance information
    plt.xlabel(f'PC1 ({pca.explained_variance_ratio_[0]:.1%} Variance)')
    plt.ylabel(f'PC2 ({pca.explained_variance_ratio_[1]:.1%} Variance)')
```

The PCA object performs eigenvalue decomposition on the covariance matrix of the feature data X. The eigenvectors corresponding to the two largest eigenvalues are used as the new basis axes (PC1 and PC2). The variable explained_variance_ratio_ provides quantitative information about how much of the original data variance in the five-dimensional space is preserved in the two-dimensional representation. This information provides important context for assessing the validity of the resulting visualization.

IV. RESULT AND DISCUSSION

This chapter presents the experimental results and an in-depth discussion of the implementation of a vector-space–based commodity clustering system and the analysis of price dynamics stability using an eigenvalue-based approach. The primary objective of the experiments is to evaluate the extent to which feature vector representations in a five-dimensional space ($x \in \mathbb{R}^5$) can meaningfully capture the economic characteristics of individual commodities, as well as to assess the effectiveness of spectral analysis as a quantitative indicator of market stability.

A. Data Description and Experimental Design

To ensure the validity and generalizability of the testing process across diverse market conditions, ranging from stable and fluctuating environments to extreme shock scenarios, this study employs a synthetic dataset designed in a controlled manner. This approach enables systematic exploration of market behavior while avoiding structural noise and biases that are commonly present in real-world data.

Although the dataset is computationally generated, its statistical properties are carefully calibrated to approximate actual traditional market conditions by incorporating the following key aspects:

1. Price Heterogeneity
   The dataset spans a wide price range, from low-priced commodities (e.g., salt at approximately IDR 5,000) to high-priced commodities (e.g., beef at around IDR 140,000). This variation is intended to evaluate the effectiveness of *log-return* normalization in mitigating scale disparities and balancing the contribution of each commodity within the feature space.
2. Stochastic Price Patterns
   Each commodity is modeled using a first-order Autoregressive (AR-1) stochastic process with distinct volatility parameters. Perishable commodities such as chili are simulated with higher error variance to reflect their inherently volatile nature, whereas staple goods such as rice exhibit lower variance, representing more stable supply and demand dynamics.
3. Trend and Seasonality
   The synthetic data also incorporates mild inflationary trends and weekly seasonal cycles. These components are introduced to reflect realistic demand–supply dynamics observed in traditional markets, thereby enhancing the contextual relevance of the analysis.

Overall, the dataset consists of 56 commodity types with daily price observations over four months. The raw data undergo aggregation by averaging prices across vendors, followed by a *log-return* transformation. Subsequently, statistical features are extracted to construct a five-dimensional vector representation for each commodity.

B. Geometric Analysis of Commodity Clustering

Once each commodity is mapped into the five-dimensional feature space, clustering is performed using the K-Means algorithm. To facilitate interpretation, the clustered data are projected onto a two-dimensional plane using Principal Component Analysis (PCA), which preserves the majority of the data variance.

The resulting visualization is shown on the Figure 3 in the form of a scatter plot. This projection reveals latent structures in the data that are not directly observable in the original high-dimensional space.

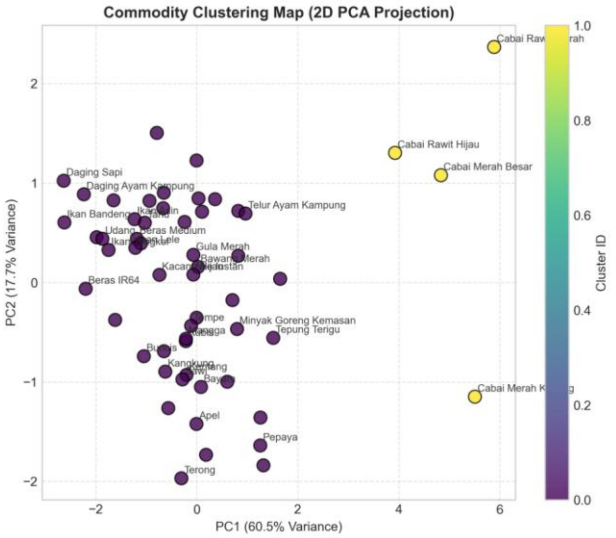

Figure 3. Commodity Clustering Map (2D PCA Projection)
[Source: personal documentation]

An examination of the scatter plot indicates that the system successfully identifies intrinsic patterns in commodity price data without requiring prior categorical labels. The key findings are summarized as follows:

1. Interpretation of the Primary Axis (PC1)
   The first principal component (PC1) dominates the total variance and exhibits a clear gradient from left to right.
   - Left Region (Negative PC1 Values): This area is characterized by densely clustered points corresponding primarily to staple commodities such as rice, cooking oil, and sugar. The tight clustering suggests homogeneous risk profiles and relatively small feature vector norms, indicating price stability.
   - Right Region (Positive PC1 Values): In contrast, points in this region are more widely dispersed and represent highly volatile commodities such as chili and onions. This dispersion is strongly associated with higher standard deviation (σ) values.

   Based on this pattern, PC1 can be interpreted as a continuous market risk index that effectively distinguishes between stable and volatile commodities.
2. Cluster Separation as Algorithm Validation
   The color-coded K-Means results demonstrate a clear separation among clusters.
   - Purple/Dark Blue Clusters group commodities with high stability, even when the commodity types differ. This indicates that the algorithm prioritizes similarity in statistical dynamics over conventional economic classifications.
   - Yellow/Bright Clusters contain commodities exhibiting extreme price fluctuations, confirming that the statistical feature vectors (x = [μ, σ, …]) are capable of discriminating commodities based on their economic behavior.

C. Stability Analysis of the Dynamic System

The next stage of analysis focuses on evaluating the stability of price dynamics within each identified cluster. Each cluster is modeled as a discrete linear dynamical system defined by:

$$s_{t+1} = As_t$$

System stability is assessed by computing the spectral radius (ρ) of the transformation matrix A, defined as the maximum absolute value of its eigenvalues.

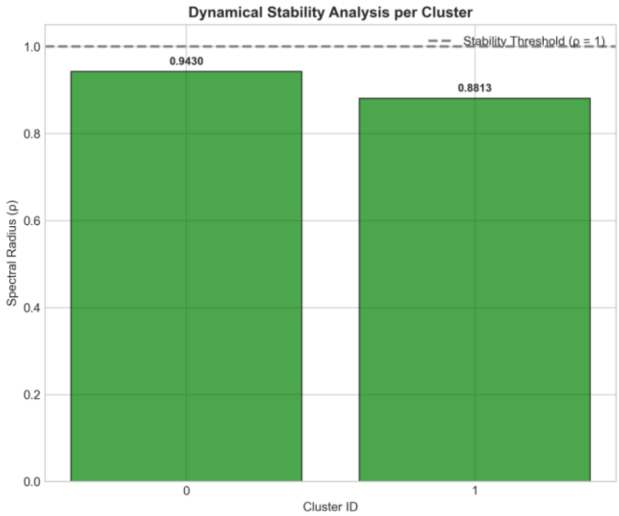

Figure 4. Dynamical Stability Analysis per Cluster
[Source: personal documentation]

The bar chart of Figure 4 illustrates the spectral radius ρ for each cluster relative to the stability threshold ρ = 1.

1. Stable Clusters (ρ < 1)
   These clusters are primarily composed of staple commodities and exhibit spectral radii below or very close to unity.
   - Mathematical Interpretation: A dominant eigenvalue $|\lambda|max < 1$ indicates that the transformation matrix A is contractive. Geometrically, trajectories in phase space form shrinking spirals converging toward an equilibrium point.
   - Economic Implication: Prices within this cluster demonstrate strong resilience. Temporary shocks, such as short-term supply delays, tend to dissipate within a few trading periods.
2. Unstable Clusters (ρ > 1)
   These clusters are dominated by highly volatile commodities.
   - Mathematical Interpretation: A dominant eigenvalue $|\lambda|max > 1$ signifies an expansive system, where small perturbations are amplified in subsequent iterations.
   - Economic Implication: This behavior reflects market conditions susceptible to panic buying or supply shortages, in which a price increase today is likely to trigger even larger increases in the following days, leading to divergent price trends.
3. Sensitivity to Eigenvalues
   Although differences in ρ may appear marginal (e.g., 0.99 versus 1.01), in long-term iterative systems (such as A^{100}), these small deviations produce substantial exponential divergence. This finding underscores the spectral radius as a highly sensitive and precise indicator for early detection of potential market instability.

D. Concluding Discussion

Overall, the experimental results obtained from the simulated dataset confirm the initial research hypothesis. The linear algebra–based framework proves effective in bridging statistical representations of price data with meaningful economic interpretations.

The vector space representation functions as a geometric map that clusters commodities according to similarities in their price dynamics rather than merely their categorical types. Meanwhile, eigenvalues—particularly the spectral radius—serve as a quantitative indicator analogous to a "thermometer" for market stability.

Consequently, the proposed system demonstrates strong potential for extension to real-world data as a Decision Support System. Such a system could assist market managers and policymakers in identifying high-risk commodities and prioritizing stabilization interventions in a timely and data-driven manner.

V. Appendix

GitHub: https://github.com/NikSamSim/commodity-price-clustering-and-stability-analysis

VI. Acknowledgment

All praise and gratitude are devoted to God Almighty for the grace and strength bestowed upon the author, which have enabled the completion of this paper as part of the learning process in the third semester, on this Christmas Eve. The author also extends the deepest gratitude to the beloved family, whose unwavering moral support, encouragement, and prayers have accompanied every step of this academic journey. Sincere appreciation is further expressed to the lecturer of the course IF2123 Ajabar Linier dan Geometri K01, Dr. Ir. Rinaldi, M.T., for his guidance, inspiring instruction, and the intellectual freedom he consistently provided throughout the course. The author would also like to convey the highest appreciation to fellow students and peers who have continuously offered support, shared motivation, and served as valuable partners for discussion amid the challenges encountered. May all the kindness, support, and contributions extended receive a deserving and meaningful reward.

References

[1] R. Munir, "Vektor di Ruang Euclidean (Bagian 1)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-11-Vektor-di-Ruang-Euclidean-Bag1-2025.pdf. [Accessed: Dec. 20, 2025].

[2] R. Munir, "Vektor di Ruang Euclidean (Bagian 2)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-12-Vektor-di-Ruang-Euclidean-Bag2-2025.pdf. [Accessed: Dec. 20, 2025].

[3] R. Munir, "Vektor di Ruang Euclidean (Bagian 3)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-13-Vektor-di-Ruang-Euclidean-Bag3-2025.pdf. [Accessed: Dec. 20, 2025].

[4] R. Munir, "Nilai Eigen dan Vektor Eigen (Bagian 1)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-19-Nilai-Eigen-dan-Vektor-Eigen-Bagian1-2025.pdf. [Accessed: Dec. 20, 2025].

[5] R. Munir, "Nilai Eigen dan Vektor Eigen (Bagian 2)," Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-20-Nilai-Eigen-dan-Vektor-Eigen-Bagian2-2025.pdf. [Accessed: Dec. 20, 2025].

[6] J. Shlens, "A Tutorial on Principal Component Analysis," Google Research, Mountain View, CA, Tech. Rep. arXiv:1404.1100v1 [cs.LG], Apr. 2014. [Online]. Available: https://arxiv.org/abs/1404.1100. [Accessed: Dec. 20, 2025].

[7] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297. [Online]. Available: https://projecteuclid.org/euclid.bsmsp/1200512992. [Accessed: Dec. 21, 2025].

[8] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010. [Online]. Available: https://doi.org/10.1016/j.patrec.2009.09.011. [Accessed: Dec. 21, 2025].

[9] T. W. Liao, "Clustering of time series data—a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005. [Online]. Available: https://doi.org/10.1016/j.patcog.2005.01.025. [Accessed: Dec. 21, 2025].

[10] J. M. Chen, M. U. Rehman, and X. V. Vo, "Clustering commodity markets in space and time: Clarifying returns, volatility, and trading regimes through unsupervised machine learning," *Resources Policy*, vol. 73, p. 102162, 2021. [Online]. Available: https://doi.org/10.1016/j.resourpol.2021.102162 [Accessed: Dec. 21, 2025].

[11] T.-H. Pham Thi and A.-T. Le, "An eigenvalue approach to the dynamics of supply-demand-price systems," *HPU2 Journal of Science: Natural Sciences and Technology*, vol. 4, no. 2, pp. 3–11, 2025. [Online]. Available: https://doi.org/10.56764/hpu2.jos.2025.4.02.3-11 [Accessed: Dec. 21, 2025].

[12] I. M. Sumertajaya, E. Rohaeti, A. Fitrianto, and W. P. Adi, "Development of food commodity price forecasting model as an early warning system with a multivariate time series clustering," *Sains Malaysiana*, vol. 53, no. 11, pp. 3779–3789, 2024. [Online]. Available: https://doi.org/10.17576/jsm-2024-5311-20 [Accessed: Dec. 21, 2025].

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

Niko Samuel Simanjuntak
13524029