# Comparison of Several Matrix Decomposition Methods in Determining Spectral Data Composition

Najwa Kahani Fatima - 13523043[1]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
[1]*najwanewa1010@gmail.com, 13523043@std.stei.itb.ac.id*

***Abstract*—Matrix is often used as a tool to analyze astronomical data, including spectrum data obtained from telescope. The received data contains information that is needed to be extracted; thus, the desired insights are able to be analyzed. In order to perform the extraction of spectral data composition, matrix decomposition needs to be performed. There are several methods that can be used in matrix decomposition. However, every method has its own pros and cons in extracting the components. This paper will compare three matrix decomposition techniques: Singular Value Decomposition, Non-negative Matrix Factorization, and Independent Component Analysis to decompose spectral data from Sloan Digital Sky Survey (SDSS).**

***Keywords*—Matrix Decomposition, Spectral Data, Data Composition.**

## I. Introduction

Mathematics, as the fundamental science of the universe, have led human to a more advanced civilization. For instance, the computers are developed from the principals in mathematics. These technologies have helped human to gain faster development of science.

One of the applications are in the field of physics and astronomy. The integration of physics, mathematics, and computers has given more opportunities in exploring the outer space and the origin of the universe. Since most objects in the outer space are unreachable by human, instruments such as telescope become a big help. Moreover, the invention of space telescopes, such as Hubble Space Telescope and James Webb Space Telescope, as well as the feasibility of performing multiwavelength observation expand the probability to discover more insights from further objects.

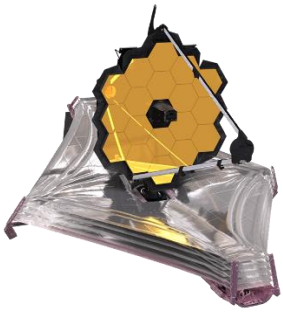

Figure 1. James Webb Space Telescope (retrieved from www.jwst.nasa.gov)

These telescopes would provide images to be processed and analyzed by human. The images are processed through a series of methods. As image is a matrix of pixels, the matrix concept in mathematics plays an important role here. With the technology of computers, image processing, which basically is a matrix processing, becomes more accurate and faster.

Data taken from images through telescope are raw data. The images could have noises or signals from the earth atmosphere, interstellar dust, or natural phenomenon such as syzygy. It might not show the targeted insight. In order to solve this, processing the matrix of pixels is really important.

A matrix of pixels could be extracted to get the meaningful features from the complex signals. However, there are numerous methods in matrix decomposition which needs to be compared. The program in computer could make this process faster. Therefore, experiment using program is also conducted.

## II. Matrix

### A. Matrices

Reference [1] defined matrix as a rectangular array of numbers which called the entries. The size of matrix is declared in terms of m $\times$ $n$ with $m$ shows the number of rows (horizontal lines) and $n$ shows the number of columns (vertical lines).

The entry that is located in row $i$ and column $j$ of matrix $A$ is donated by $a_{ij}$. Therefore, the general $m \times n$ matrix $A$ is written as following.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

A matrix $A$ which has $n \times n$ size is called a square matrix of order $n$. The entries $a_{11}, a_{22}, \ldots, a_{nn}$ are on the main diagonal of $A$.

Two matrices are equal if they have the same size and the corresponding entries are equal. That means, $A$ and $B$ are equal if and only if the number of rows of $A$ is the same with the number of rows of $B$ and the number of columns of $A$ is the same with the number of columns of $B$. In all corresponding values of $i$ and $j$, the below expression must also be true.

$$a_{ij} = b_{ij}$$

If $A$ and $B$ are matrices with the same size, then the sum and

the difference of $A$ and $B$ is obtained by adding or subtracting the entries, such as below.

$$(A \pm B)_{ij} = a_{ij} \pm b_{ij}$$

To get the product of matrix $A$ with the scalar $c$, all entries in $A$ will be multiplied by $c$. Meanwhile, the product of two matrices $A$ and $B$ is only valid if the size of $A$ is $m \times k$ and the size of $B$ is $k \times n$. This product will result in a matrix $C$ with size $m \times n$. The entries in $C$ can be calculated from the equation below.

$$C_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}$$

Matrices also often declared as a linear combination. If $A_1, A_2, \ldots, A_n$ are matrices of the same size and $c_1, c_2, \ldots, c_n$ are scalars, then $c_1A_1 + c_2A_2 + \cdots + c_nA_n$ is the linear combination of $A_1, A_2, \ldots, A_n$ with coefficient $c_1, c_2, \ldots, c_r$. This also works for matrix product with linear combinations. If $A$ is $m \times n$ matrix and $\vec{x}$ an $n \times 1$ column vector as below,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \text{ and } \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

then the product of $A$ and $\vec{x}$ is a matrix $C$ with size $m \times 1$ as following.

$$C = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

If $A$ is an $m \times n$ matrix, then the matrix $A^T$ is the transpose of $A$ and has a size of $n \times m$. The entries in $A^T$ corresponds to the entries in $A$ by interchanging the rows and columns of $A$. Suppose $B = A^T$, the entry $b_{ij}$ equals to the entry $a_{ji}$ in matrix $A$.

A zero matrix is a matrix whose entries are all zeros. In another hand, the identity matrix $I$ is a square matrix with all 1's on the main diagonal and zeroes elsewhere. Any matrix $A$ if multiplied by $I$ will results in the matrix $A$ itself. Below is the example of an identity matrix with size $3 \times 3$.

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If $A$ and $B$ are square matrices with same size and they satisfy $AB = BA = I$, then $B$ is the inverse of $A$ and $A$ is nonsingular (invertible). $B$ can be written as $A^{-1}$. If there is no $B$ that satisfies the expression, then $A$ is singular.

For matrix $A$ with size $2 \times 2$,

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

the inverse of $A$ exists if only if the determinant is not zero ($\det(A) = ad - bc \neq 0$). The inverse of $A^{-1}$ is declared as below.

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

In general, for any matrix $A$ with the determinant not equals to zero, the inverse is calculated as

$$A^{-1} = \frac{1}{\det(A)} adj(A)$$

with $adj(A)$ is the adjoin of $A$, which is the transpose of cofactor matrix of $A$. A cofactor matrix is the matrix generated from the cofactor of each entry in A. The cofactor at row $i$ and column $j$ is $C_{ij} = (-1)^{i+j}M_{ij}$, with $M_{ij}$ is the minor of the corresponding entry. The minor $M_{ij}$ is the determinant of the submatrix that exclude the $i$-th row and the $j$-th column.

A diagonal matrix is a square matrix in which all the entries off the main diagonal is zero. Below is the example of diagonal matrices.

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & -10 \end{bmatrix}, \begin{bmatrix} 8 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

A square matrix which all entries above the main diagonal are zero is called lower triangular matrix. If all entries below the main diagonal are zero, it is called an upper triangular matrix. Here, $A$ is a lower triangular matrix and $B$ is an upper triangular matrix.

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, B = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

### *B. Eigen Value and Eigen Vector*

An eigen vector of $A$ $(n \times n)$ is a nonzero vector $\vec{x}$ in $R^n$ if $A\vec{x}$ is a scalar multiple of $\vec{x}$. This could be denoted as below with $\lambda$ is any scalar. The $\lambda$ is called the eigen value of $A$ and $\vec{x}$ is the eigen vector to $\lambda$.

$$A\vec{x} = \lambda\vec{x}$$

This shows that the equation above compresses or stretches eigen vector $\vec{x}$ by a factor of $\lambda$.

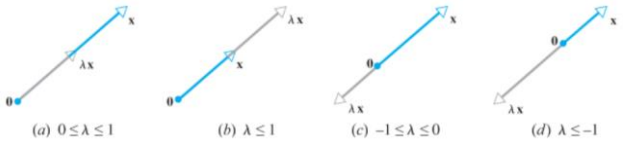


Figure 2. The eigen value $\lambda$ compresses or stretches $\vec{x}$ based on the range of value (retrieved from [1]).

The eigen values of matrix $A$ can be computed from the equation below which must have a nonzero solution for $\vec{x}$. In order to satisfy that, the coefficient matrix must have a zero

determinant. The equation below is called the characteristic equation of $A$.

$$\det(\lambda I - A)\,\vec{x} = 0$$

## III. Matrix Decomposition

There are several methods to decompose a matrix. In this section, four different methods will be discussed.

### A. Singular Value Decomposition

A symmetric matrix $A$ can be expressed as $A = PDP^T$ with $P$ is an orthogonal matrix of eigen vectors of $A$ and $D$ is the diagonal matrix which entries the eigen values of $A$. If $n \times n$ is not symmetric, it has Hessenberg decomposition $A = PHP^T$ and if $A$ has real eigen values, then it has Schur decomposition $A = PSP^T$. In another hand, a general square matrix decomposition form is $A = PDP^{-1}$ or one might be in the form of $A = U\Sigma V^T$ [1].

In singular value decomposition, the target factorization is the second one, which is

$$A = U\Sigma V^T$$

where Σ is a diagonal matrix with entries of singular value ($\sigma$) of $A$, which value is $\sigma_n = \sqrt{\lambda_n}$ and $\lambda$ is the eigen value of $A^TA$. Here, the $V$ matrix consist of normalized eigen vectors ($\vec{v}_n$) of $A^TA$ as its column vectors and $U$ consists of $\overrightarrow{u_n} = \frac{1}{\sigma_n} A\overrightarrow{v_n}$ as its column vectors. Vectors in $U$ can also be computed from the eigen vectors of $AA^T$.

### B. Non-Negative Matrix Factorization

Non-Negative Matrix Factorization only create positive factors from the original matrix. Since it only creates positive factors, it creates a constraint in certain applications such as in pixels and probability that cannot be negative [3]. This method usually be used in image processing, transcription process, cryptic encoding and decoding, music and videos decomposition, and text mining [4].

The goal of NMF is to decompose a matrix $X$ into two matrices as below with all values are positive [5].

$$X \approx WH$$

Here, $X$ has dimension $d \times n$ with $d$ dimensional data and $n$ individual data vectors. $W$ has the size of $d \times q$ and $H$ has the size of $q \times n$. The $q$ is the desired number of templates to fit. The decomposition started by setting random positive number in $W$ and $H$.

In order to find the best corresponding entry, a new approach to update the entries is introduced to minimize the error as follows [6].

$$H \leftarrow H \cdot \frac{W^TX}{W^TWH}$$
$$W \leftarrow W \cdot \frac{XH^T}{WHH^T}$$

### C. Independent Component Analysis

Independent Component Analysis (ICA) is a technique to separate independent sources from mixed signals. It assumes that hidden independent components in a mixed signal are independent and are non-Gaussian [7].

ICA process consists of preprocessing, ICA estimation, and extracting independent components. Preprocessing includes data centering and whitening to decorrelates the variables and ensures they have unit variance. ICA estimation includes initializing a matrix, supposed $W$, the maximization of Non-Gaussianity using FastICA. Extracting independent components is denoted as below with $S$ is the matrix with independent components once the $W$ converges.

$$S = WX_{whitened}$$

### D. Principal Component Analysis

Principal Component Analysis (PCA) is not a matrix decomposition method itself, yet a method to find the most significant features in a matrix. Matrix decomposition technique such as SVD is used in one of the processes of finding PCA. It is a statistical technique for identifying data by linear mapping. It transforms a number of correlated variables into principal components that consist of a smaller number of uncorrelated variables. The transformed data has most of the variation in the first few components, the remaining components can be ignored. The steps to transform the data using PCA consists of five steps as following [2].

1. Subtracting the mean from each data dimension that creates the centered data set with mean zero ($A_\mu$).
2. Calculating the covariance matrix of the centered data.
3. Calculating the eigen vectors and the eigen values of the covariance matrix. Here, the eigen value will represent the variance within a given component.
4. Ordering the eigen vector based on the eigen values decrementally (from highest to lowest).
5. Forming feature vector by taking the retained eigen vectors which are arranged column wise. The feature vectors would be as below.

$$FV = Feature\ Vector = (ev_1\ \ ev_2\ \ \dots\ \ ev_n)$$

6. Transposing feature vector and subtracting its mean from the original centered data. This results in obtaining the PCA data. Thus, the PCA data is calculated as follows:

$$PCA_{data} = FV^T \times A_\mu$$

## IV. Spectral Data

Spectral data are data that is related to spectrum of frequencies. It refers to information that is obtained from diverse spectroscopic techniques [8]. In this paper, the spectral data will focus on astronomical data.

In observational astronomy, spectroscopy is bringing huge impact and importance. It is the major key of understanding the physics and chemical properties of cosmic objects, including galaxies, stars, nebulae, and others. The concept of spectroscopy

originated from the theory of diffractive element by Isaac Newton, saying that light can be decomposed into contributions of different wavelength [9].

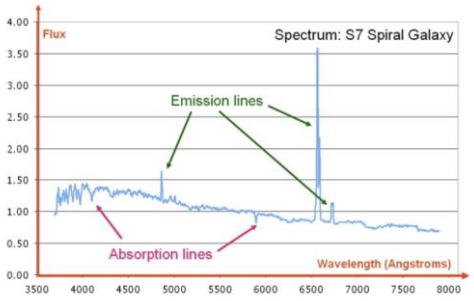


Figure 3. Example of spectral data: Spiral Galaxy S7 spectrum (Retrieved from [9]).

The spectral data received directly from observational instruments contains many noises and signals that come from the earth atmosphere, interstellar dust, the movement of the cosmic object such as redshift, and its interaction with other object near them. The data can also contain lots of information that is influenced by several features that is dominant to the data. Decomposing the data has always been one step in data processing to get the significance of certain variance that most influenced the data. This will be giving insights about more information to be analyzed.

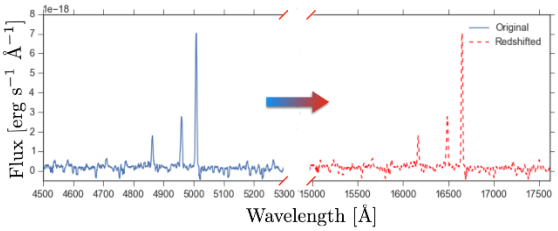


Figure 4. Redshifted star spectra data of MOSDEF galaxy (retrieved from https://mosdef.astro.berkeley.edu)

## V. Proposed Method

Determining spectral data composition in astronomy requires mathematics fundamental concept related to matrices. In this paper, the four methods of matrix decomposition using PCA, SVD, NMF, and ICA is conducted to perform spectral data decomposition of an astronomical data. These four methods will be compared to each other.

The astronomical spectra data used for experiment is taken from Sloan Digital Sky Survey (SDSS) repository, focusing on photometric and spectroscopic observations from bright objects. The data is retrieved using SQL query as follows.

```
SELECT TOP 10
p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z
,p.run, p.rerun, p.camcol, p.field,
s.specobjid, s.class, s.z as redshift,
s.plate, s.mjd, s.fiberid
FROM PhotoObj AS p
JOIN SpecObj AS s ON s.bestobjid = p.objid
WHERE
    p.u BETWEEN 0 AND 19.6
    AND g BETWEEN 0 AND 20
```

The data used in the experiment is one of the results of the query. The experiment is conducted using libraries in python. The libraries used in the experiment are numpy, astropy, matplotlib, and scikit learn. Astropy is used to load the `.fits` data from the SDSS repository. Matplotlib is used to plot the data into images. Meanwhile, numpy and scikit learn are used to perform the matrix decomposition.

## VI. Program Experiment

The data processed in the experiment is an unknown and unclassified object located at right ascension $RA$ (309.88°) and declination $\delta$ (−5.89°). The object was observed on MJD 52164 or 12 September 2001. It was observed on telescope SDSS 2.5-M. Below is the flux vs. wavelength plot of the object.

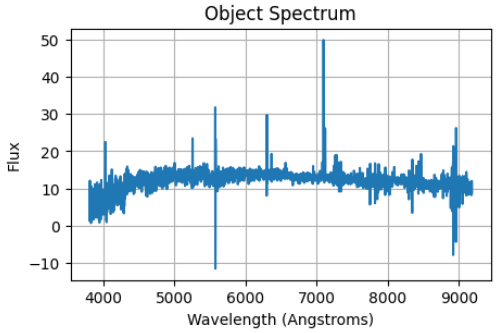


Figure 5. The plot of flux vs. wavelength of the object.

The experiment is carried out in python with four extracted components. The number four is chosen refers to the previous study which used four eigen spectra in research of quasars spectrum from SDSS dataset [10]. The program is constructed as follows.

1. Importing libraries

```
import numpy as np
from astropy.io import fits
from sklearn.decomposition import PCA, NMF,
FastICA, TruncatedSVD
import matplotlib.pyplot as plt
from sklearn.preprocessing import
StandardScaler
```

2. Loading data

```
def load_fits_data(file_path):
    with fits.open(file_path) as hdul:
        data = hdul[1].data
        flux_data = data['flux']
        return flux_data.astype(np.float64)
```

3. Creating sliding windows to create matrix from data

```
def create_sliding_window_matrix(data,
window_size=50):
    n_samples = len(data) - window_size + 1
    matrix = np.zeros((n_samples,
    window_size))
    for i in range(n_samples):
        matrix[i] = data[i:i + window_size]
    return matrix
```

4. Applying matrix decomposition

```
def apply_decompositions(data,
n_components=4):
    window_size = 50
    data_matrix =
    create_sliding_window_matrix(data,
    window_size)

    scaler = StandardScaler()
    data_scaled =
    scaler.fit_transform(data_matrix)

    decomposition_methods = {
       'PCA': PCA(n_components =
       n_components),
       'SVD': TruncatedSVD(n_components =
       n_components),
       'NMF': NMF(n_components =
       n_components, init='random',
       random_state=0),
       'ICA': FastICA(n_components =
       n_components, random_state=0)
    }

    results = {}
    for name, method in
    decomposition_methods.items():
        if name == 'NMF':
           data_nmf = data_scaled -
           data_scaled.min()
           transformed =
           method.fit_transform(data_nmf)
        else:
           transformed =
           method.fit_transform(data_scaled
           )

        results[name] = {
           'transformed': transformed,
           'components': method.components_
           if hasattr(method,
           'components_') else None,
           'explained_variance':
           method.explained_variance_ratio_
           if hasattr(method,
           'explained_variance_ratio_')
           else None
        }

    return results
```

5. Plotting components

```
def plot_components(results,
n_components=4):
    fig, axes = plt.subplots(len(results),
    1, figsize=(12, 3*len(results)))
    axes = np.array([axes]) if len(results)
    == 1 else axes

    for idx, (method_name, result) in
    enumerate(results.items()):
        components = result['components']
        if components is not None:
           for i in range min(n_components,
           len(components))):
              axes[idx].plot(components[i],
              label=f'Component {i+1}')
              axes[idx].set_title(f'{method_n
              ame} Components')
              axes[idx].legend()
              axes[idx].grid(True)
    plt.tight_layout()
    plt.show()
```

6. Main program

```
def main():
    file_path = '1-spec-0634-52164-
    0256.fits'
    data = load_fits_data(file_path)
    results = apply_decompositions(data)
    plot_components(results)

main()
```

Below is the result of the plotting of each component extracted from the data.

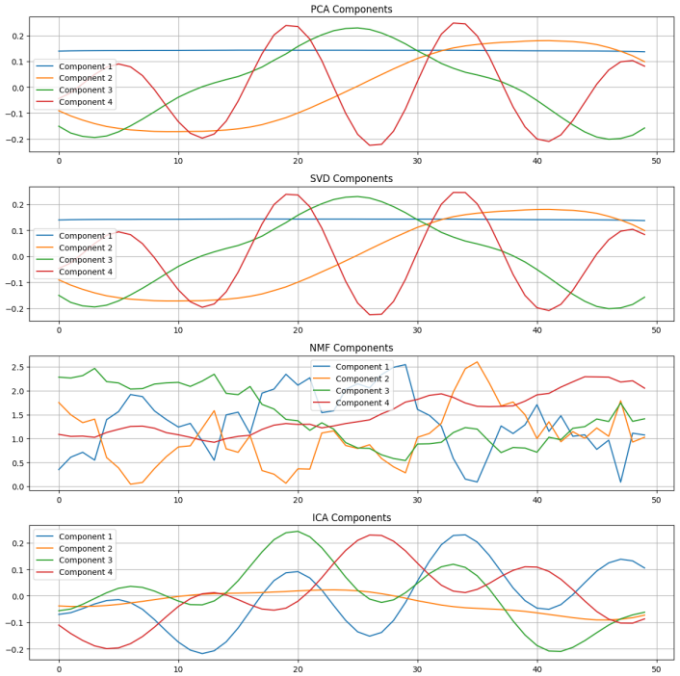


Figure 6. Matrix decomposition result using SVD, PCA (implementation of SVD), NMF, and ICA. $x$ axis represent window size/position and $y$ axis represent component coefficient.

Based on the result, it is seen that PCA and SVD has the same plot since they are mathematically relevant. They provide a smooth periodic component (2-4) and a stable baseline of component 1. The first component might represent the mean spectrum, which explains the largest variance. The smooth oscillations indicate structured patterns in the data. Each component is uncorrelated (orthogonal) to each other.

The NMF component has more erratic pattern due to non-negativity constraint. It has higher amplitude on range 0-2.5. The less smooth components shows that it captures local features. The component variation suggests different emission or absorption features.

In another hand, ICA focuses on statistical independence rather than orthogonality. It gives more emphasis on independence between components and captures statistically independent spectral features.

From the analysis, it can be concluded that PCA and SVD method is best for overall variance (matrix) decomposition. Meanwhile, NMF is useful for identifying physical spectral features since it has positive value constraints. The ICA method is better at separating independent sources in the spectrum.

## VII. Conclusion

Matrix decomposition is a method that can be used in extracting features of spectral data in astronomy. However, in order to find the best and most relevant result, not every method can be applied. PCA and SVD is optimal for data variance maximization. In another hand, NMF components match physical spectral features since it only processes positive values which match real spectral intensities and has more irregular pattern. Meanwhile, ICA focuses on statistical independence and good for separating mixed signals.

Further research that examines more sample data can be carried out to get a more comprehensive result. Other matrix decomposition method also can be experimented and compared to the one that has been experimented in this paper.

## IX. Acknowledgment

The author would like to thank to God for the guidance throughout the process of learning and writing this paper. The author would also like to deliver biggest gratitude to IF2123 Linear Geometry and Algebra lecturers for sharing and guiding the student in learning the materials throughout the semester. The author would also like to thank to family and friends who have accompanied the journey of joy and sorrow since the start of the author's university journey.

## References

[1] Anton, H. (1994) Elementary Linear Algebra. 7th Edition, John Wiley & Sons, Hoboken.
[2] Gandhi, Vaibhav. (2015). Brain-Computer Interfacing for Assistive Robotics.
[3] Theodoridis, Sergios. (2020). Machine Learning: A Bayesian and Optimization Perspective (Second Edition).
[4] Belyadi, H., Haghighat, A. (2021). Machine Learning Guide for Oil and Gas Using Python.
[5] Green, D., Bailey, S. (2024). Algorithms for Non-Negative Matrix Factorization on Noisy Data With Negative Values.
[6] Lee, D., Seung, H.S. (2000). "Algorithms for Non-negative Matrix Factorization," in Advances in Neural Information Processing Systems.
[7] Choudhury A. D., Pal, Arpan. (2022). "Sensor Signal Analytics," in New Frontiers of Cardiovascular Screening Using Unobtrusive Sensors, AI, and IoT.
[8] Significance of Spectral data. Retrieved from https://www.wisdomlib.org/concept/spectral-data.
[9] Kaminski, A., Heidt, J., Pfeifer, V. (2021). Spectroscopy - Data Reduction and Interpretation.
[10] Yip, C. W., Connolly, A. J., Berk, D. E. V., Ma, Z., Frieman, J. A., Subbarao, M., Szalay, A. S., Richards, G. T., Hall, P. B., Schneider, D. P., Hopkinds, A. M., Trump, J., Brinkmann, J. (2004). Spectral Classification of Quasars in The Sloan Digital Sky Survey: Eigenspectra, Redshift, And Luminosity Effects.

## Statement

I hereby declare that the paper I wrote is my own writing, not an adaptation, or translation of someone else's paper, and not plagiarized.

Bandung, 31 Desember 2024

Najwa Kahani Fatima - 13523043