Circos Graph Application to Represent Similarity Between Majors Curriculum in ITB Undergraduate Programs

Akbar Maulana Ridho - 13521093[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13521093@std.stei.itb.ac.id

***Abstract*—Circos graph is a type of weighted graph representation in a circular layout. This paper tries to represent undergraduate majors' curriculum similarity by using a weighted graph with a node as the major and an edge as the similarity weight. The data was collected from the ITB Student Website's (SIX) curriculum page. The similarity was calculated by taking keywords from each subject in each major and then calculating similarity from each pair of the major aggregated keyword. We use Spacy's pre-trained NLP model to extract keywords and calculate document similarity. As result, we were able to make a Circos graph that was able to represent curriculum similarity for each undergraduate major in ITB.**

***Keywords*—Circos graph, Weighted graph, Curriculum similarity, Natural Language Processing**

I. INTRODUCTION

In this digital world, data is everything. However, most of the data are raw and need to be processed further and extract meaningful information from it. After that, visualization can be used to help us comprehend data in a better way.

Circos is a type of visualization that was originally used for visualizing alignments and structural variation in genomic data. After Circos' popularity grow, this type of visualization began used to visualize another type of data [1].

On the other hand, a weighted graph is a type of graph that have weights on its edge. This type of graph can represent entity relation and its relation significance. This type of graph has several usages, such as the shortest path problem, traveling salesman problem, and weighted network.

This topic was chosen out of the author's curiosity about how related each major curriculum is to each other. While it is common for each student to know their major relation with another major in their faculty, it is still helpful to know every major relationship in general.

This led us to a question on how we calculate the similarity between every major. While it is difficult to collect and measure similarity based on the student or lecturer's view, it also potentially has a big subjectivity. Thus, we decided to collect every subject's syllabus in every major curriculum. The data was collected from ITB Student's Site (SIX).

After collecting the data, we still need to clean, and process scraped data to extract the desired information. Based on quick research, we decided to use several Natural Language Processing methods to extract keywords from the syllabus and calculate the similarity between the major's aggregated keywords. This was done using the ready-to-use Spacy NLP library and its pre-trained English language model. Before being visualized into a Circos graph, the relation was first represented as a weighted graph.

While this method may not be the best to calculate similarity due to the author's lack of knowledge about NLP, we will use the best method available to us to maximize the result.

II. THEORETICAL BASIS

A. Graph

A graph is used to represent discrete objects and the relation of each object. In definition, graph $G = (V, E)$ where $V = \{v_1, v_2, \dots, v_n\}$ that are a set of vertices and $E = \{e_1, e_2, \dots e_n\}$ that are a set of edges [2].

Based on loop existence, the graph is divided into two categories.

1. Simple graph. This graph type of graph doesn't contain a loop or multiple edges.
2. Unsimple graph. This type of graph contains a loop or multiple edges.

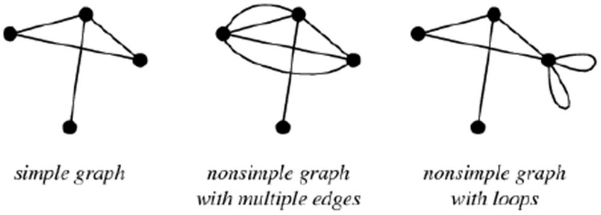

simple graph
nonsimple graph with multiple edges
nonsimple graph with loops

Image 1 Type of graph
(source: Rinaldi Munir/ IF2120 Discrete Mathematics)

Based on edge direction orientation, the graph is also divided into two categories.

1. Undirected graph. This type of graph does not have direction orientation.
2. Directed graph. This type of graph has directions on each edge.

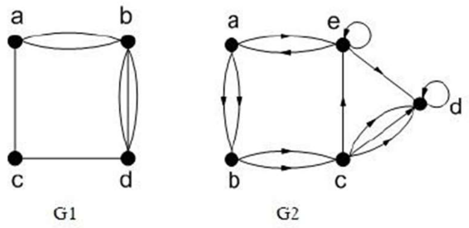


Image 2 G1 Undirected Graph, G2 Directed Graph
(source: Rinaldi Munir/ IF2120 Discrete Mathematics)

There are many terminologies of a graph, some of them are:

1. Adjacency. Two vertexes are adjacent if two of them are directly connected.
2. Incidence. For every edge $e = (v_j, v_k)$, e is incidence if e incidence with vertex v_k or vertex v_j.
3. Degree. The degree of a vertex is the total of edges that are incident with the vertex.
4. Weighted graph. A weighted graph is a graph where each of its edges has a value/ weight.

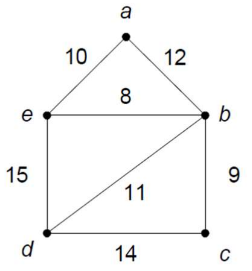
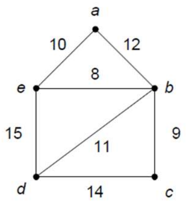


Image 3 Weighted Graph
(source: Rinaldi Munir/ IF2120 Discrete Mathematics)

There are many ways to represent a graph, some of them are adjacency matrix and adjacency list [9].

For the adjacency matrix, we define $A = [a_{ij}]$ so that $a_{ij} = 1$ if vertex i and j are adjacent and $a_{ij} = 0$ if both are not. However, a graph is a weighted matrix, $a_{ij} = w_{ij}$ if vertex i and j are adjacent and w_{ij} is the e_{ij} weight. If vertex i and j are not adjacent, $a_{ij} = \infty$.



$$\begin{array}{c|ccccc|} & a & b & c & d & e \\ a & \infty & 12 & \infty & \infty & 10 \\ b & 12 & \infty & 9 & 11 & 8 \\ c & \infty & 9 & \infty & 14 & \infty \\ d & \infty & 11 & 14 & \infty & 15 \\ e & 10 & 8 & \infty & 15 & \infty \end{array}$$

Image 4 An example of a weighted graph represented in an adjacency matrix
(source: Rinaldi Munir/ IF2120 Discrete Mathematics)

For the adjacency list, we create a list of vertexes. Each vertex has a property that holds a list of another vertex that is adjacent to its vertex. As for the weighted graph, we could add a weight property on a list that we previously described.

B. Circos Graph

Circos is software for visualizing data and information that visualize data in a circular layout. This type of visualization is ideal for exploring the relationship between object and position [1].

While Circos was first developed to display genomics data, it does not mean Circos doesn't work for other types of data. Circos is flexible and could be used to represent other types of data as well.

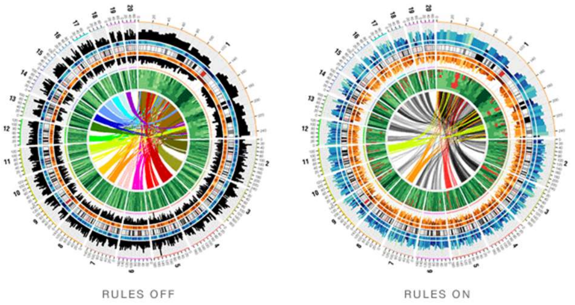


Image 5 An Example of Circos Visualization
(Source: http://circos.ca/img/circos-rules.png)

C. Natural Language Processing

Natural Language Processing (NLP) is a branch of computer science that is concerned to give a computer the ability to understand text and spoken words the same way humans can do [3].

NLP combines computational linguistics with statistical and machine learning models. These technologies enable a computer to process human language and understand its full meaning.

NLP tasks that are used in this paper are:

1. Part of speech tagging is a process of determining the part of speech of a particular word or piece of text based on its use and context.
2. Named entity recognition (NER), is a process of identifying words or phrases that are useful entities.
3. Stop words removal. Stop words are a list of high-frequency words, such as the, to, also, etc. that we usually want to remove from the document before further processing. Stop words could be safely removed since it has little significance lexically [4].
4. Chunking is a process of segments and labels in multi-token sequences. Noun phrase chunking is a process of chunking that search for chunks corresponding to individual noun phrases [4].

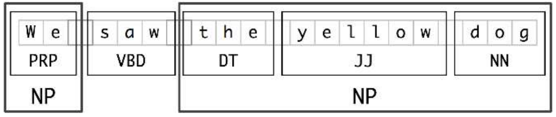


Image 6 Segmentation and Labeling at Both the Token and Chunk Levels
(Source: https://www.nltk.org/book/ch07.html)

5. Lemmatization is a process of vocabulary and morphological analysis of words, normally aiming to remove inflectional endings and to convert to the base or dictionary form of a word [5].
6. Word2vec is an algorithm that uses a neural network model to learn word associations from a large corpus of text [6]. This algorithm is used to calculate the similarity between two documents.

D. Min-Max Normalization

Min-Max normalization is used to scale a set of numbers so that every number value will be inside a range of 0 and 1. Normalization was done because variables that are measured at different scales don't necessarily contribute equally to the model and could lead to bias [7]. Here is the mathematical formula for min-max normalization.

$$x_{scaled} = \frac{x - x_{minimum}}{x_{maximum} - x_{minimum}}$$

E. Percentile

Percentile is a measure in statistics indicating the value below which a given percentage of observation in a group of observations falls [8]. Quantile, on the other hand, is a percentile with the percentage value represented in a decimal value.

III. TOOLS AND METHODOLOGY

A. Tools

These are the tools and libraries that are used in this paper:

1. Python 3.10.
2. Jupyter Notebook.
3. Selenium, to automate website scraping tasks.
4. Beautiful soup, to parse scraped HTML pages.
5. Spacy, ready-to-use NLP tools, and pre-trained models. For this paper, we use Spacy en_code_web_lg pre-trained models. Here is an overview of Spacy library architecture and model pipeline.

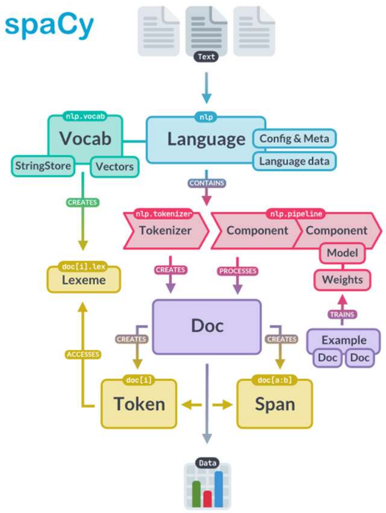


Image 7 Spacy Library Architecture
(source: https://spacy.io/api)

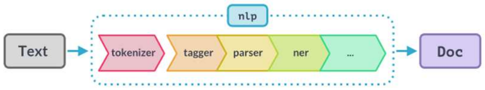


Image 8 Spacy model pipeline
(source: https://spacy.io/api)

6. NetworkX is a library that can represent and manipulate complex graph networks.
7. Numpy for easy array and data manipulation.
8. Matplotlib for easier graph plotting
9. Nxviz, a network visualization tool for NetworkX. We use this module to draw the Circos graph.
10. Pandas

B. Data Scraping

For scraping, we use Selenium and Chrome Web Driver to automate actions. Here is a general step on how the data was scraped from SIX:

1. From the controlled web browser, log in to SIX.
2. Open the major's curriculum page for every combination of faculty and major's code.
3. In the major's curriculum page, save the page as an HTML file then scan every compulsory and elective subject page.
4. Save every subject page as an HTML file.

We do not take specific data in this step, instead, we scrape the whole pages. This enables us to extract different parts of information if needed while requiring us to re-scrape the site.

C. Data Processing

Data processing was divided into several steps. The first one was to extract desired information from previously saved data. These are the steps.

1. For every major page, iterate every subject that is not in subject exclusion.
2. For every subject, parse the HTML files with beautiful soup and extract the English version of the subject's name and syllabus.
3. Remove numbering patterns, unnecessary whitespace, and encoding errors with regular expressions.
4. Convert the syllabus into lowercase.
5. Feed syllabus text into Spacy NLP model then extracts noun chunks that have more than one word.
6. Noun phrases that were previously extracted are then converted into lemmatized versions and their stop words are removed.
7. In the end, we will get a list of majors that contain the list of subjects. Each subject contains a subject name and a list of extracted keywords from the syllabus.

Notice that we exclude some subjects and only use multi-word noun chunks. This step was chosen to reduce common similarities. Some subjects like English, Indonesian Writing, Final Project, etc. are removed since they are compulsory in each major and not a unique subjects in that major. Then, we exclude single-word noun chunks since it increases the possibility of keywords having many similarities while doesn't tell whether it really similar or they only have common words.

Also, we use the English version of the subject name and syllabus due to the limited amount of NLP resources and pre-trained NLP models in the Indonesian language.

After this step, we could start to compare the similarity between each major's curriculum. Here are the steps.

1. For every major, we combine every subject keyword into a large document of subject names and syllabus keywords.
2. For every subject, we only take the five longest keywords to include the most significant keywords. We assume that a long keyword means it contains higher importance.

3. For every pair of majors, load both documents into the Spacy NLP model and compare its similarity.
4. In the end, we will get a list of major pairs and its similarity. Every pair of major acts as an edge and every major function as a vertex. Its similarity is the weight of the edge.
5. Lastly, we do a min-max normalization and rescale the weight. Normalization was done by subtracting every weight with minimum weight and dividing it by the difference between the maximum and minimum of the weight. After that, we multiply it by 100 to have its value range from 0 to 100. Normalization processes were needed to make similarity-difference more noticeable since it has quite a high average similarity.

D. Visualization

There are several steps before we display the processed data in the Circos graph. Here are the steps.

1. Load major-pair and normalized weight. Rescale the weight into the range of 0 to 10.
2. Create a graph instance from NetworkX's Graph implementation. This graph uses adjacency list representation to represent our weighted graph.
3. Insert every major as nodes/ vertex of the graph.
4. Calculate three different cutoff values for weight, such as base cutoff (85% percentile or 0.85 quantiles), mid cutoff (95% percentile or 0.95 quantiles), and high cutoff (98% percentile or 0.98 quantiles).
5. For every edge that falls into the 85% percentile, add an edge to the graph.
6. For highlighting, every edge that weighs more than a high cutoff will have the dominant color and lowest transparency. Every edge above mid-cutoff will have less dominant color and transparency. The same goes for every edge that falls below mid-cutoff.
7. Draw Circos plot with Nxviz library.

IV. COLLECTED DATA

Before we are going into the graph result, let us a quick overview of the distribution and statistics of our data. Here are the stats on the raw data that we previously processed.

Name	**Unique Page Count**
Major's Page	50
Compulsory Subject Page	1584
Elective Subject Page	1120
Total Page	2754

Table 1 Raw data information

After processing, here is the data distribution.

	Mean	**Min**	**Max**	**25%**	**75%**
Keywords each Subject	9.2	0	61	5	12
Keywords each Major	517	226	986	427	600
Keywords length	29.9	5	639	16	34

Table 2 Keywords count on subjects and majors

As for the keywords count itself, there is a total of 25865 keywords across the majors.

Last, here is the data distribution on our similarity calculation.

	Count	Mean	Std	Min	25%	75%	Max
Original	1225	0.94	0.03	0.8	0.92	0.96	0.99
Normalized	1225	71	15.45	0	62	82	100

Table 3 Weight distribution

Even after normalization, the similarity means still quite high. The author itself do not sure how much of it is a noise that makes similarity higher (for example words that do not mean anything but are included in the analysis) and how much of it is the real similarity.

Here is an adjacency matrix representation of a subset of the similarity data.

$$\begin{bmatrix} \infty & 82 & 66 & 96 & 62 \\ 82 & \infty & 89 & 71 & 84 \\ 66 & 89 & \infty & 54 & 71 \\ 96 & 71 & 54 & \infty & 62 \\ 62 & 84 & 71 & 62 & \infty \end{bmatrix}$$

Each index of row or column corresponds to this list of major's code $[FA, MR, DP, FK, MB]$.

V. RESULT

After some experiments, here is the final Circos graph. We previously described that only the edge that falls into the 85% percentile is included. For an edge that falls between 85% and 95% percentile we call them weakly similar and have a yellow color in the graph. For an edge that falls between 95% and 98% we call them moderately similar and have purple color in the graph. Last, for every edge that falls above the 98% percentile, we call them highly similar and have greenish color in the graph. In total, we have 1225 edges. There 122 of them fall into the weakly similar category, thirty-seven fall into the moderately similar category, and 25 falls into the highly similar category.

Based on the graph, several faculties have each of its major moderately or highly similar to each other. These faculties are FTTM, FTSL, FTMD, FTI, STEI, SF, SBM, and SITH. It is also interesting to see that a group of engineering faculties are moderately and highly similar to each other. These groups are FTTM, FTSL, FTMD, and FTI.

It is also interesting to see that some of the faculties are isolated, such as SAPPK, SBM, SF, and FSRD. For SAPPK, even though AR have weak similarity with another major, it is a similarity with KR and SR (part of FSRD) that are isolated as well. For SBM, we could assume that it is true that they are isolated and MB and MK are highly similar. For SF, even though they are isolated, they still have some moderate and weak similarities with some majors in SITH and KI.

Another noticeable relation is there are many moderate and highly similar majors between FTI and STEI. As we know, EL and IF were once a part of FTI. It is good to see that their history was still traceable in this graph.

Let's see how the degree distribution in each vertex. For this, we add two more categories. Column ">mean degree" means the vertex's degree that every weight value is more than the mean and less than 85% percentile. Column "<mean degree" means the vertex's degree that every weight value is less than

the mean.

However, it does not mean all intended relations are represented in this graph. AR and DI should have at least weakly related, KI should have at least moderate similarity with any of SF's majors, and MA should have at least weak or moderate similarity with more majors.

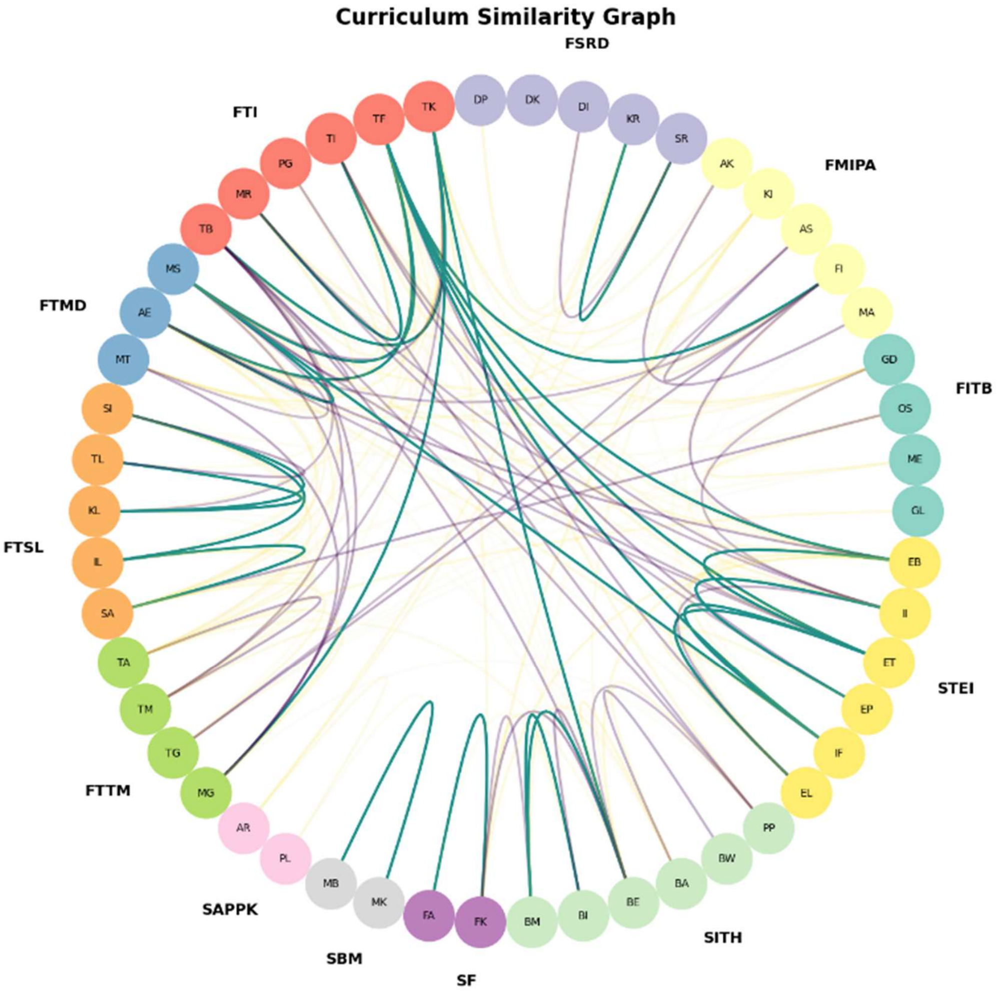


Image 9 Curriculum Similarity Graph

After seeing through the graph, let's see detailed data about vertexes (majors) degree distribution.

	Degree				
	< mean	> mean	>85%	>95%	>98%
Mean	22	20	5	1.5	1
Min	6	3	0	0	0
25%	13	19	2.25	0.25	0
50%	20	21.5	4	1	1
75%	26	24	6.75	2	1
Max	45	30	14	6	6

Table 4 Graph vertex degree distribution

Also, here is a snapshot of the degree distribution data.

	Degree count				
	<mean	>mean	>85%	>95%	>98%
TF	11	20	11	1	6
BE	6	27	10	4	2
AE	8	22	14	3	2
MS	13	21	7	5	3
TK	11	21	12	2	3
…	…	…	…	…	…
MB	41	6	1	0	1
MK	40	8	0	0	1
DK	41	6	1	1	0
KR	44	3	2	0	0
SR	45	4	0	0	0

Table 5 Top 5 and bottom five vertexes based on degree count

Based on the table, TF, BE, AE, MS, and TK are majors that have significant similarities with other majors. TF, for example, has 6 high similarities with other majors, such as MS, AE, FI, EB, ET, and IF.

On the other hand, MB, MK, DK, KR, and SR are majors that have the least significant similarity with other majors. SR, for example, only has 4 degrees of relation that are above average and below the 85% percentile. These majors are from SBM and FSRD.

VI. Conclusion

As we have seen in the previous part, the Circos graph could be used to represent the similarity between undergraduate majors in ITB. One of its advantages is that it could flatten the graph representation so that it could be represented on a 2d graph. In other words, the Circos graph can display a 3d graph connection into a 2d visualization. With this graph, we could see a more broad and more general view of how each major related to another major inside or outside of their faculties. It is also nice to see how edge existence, transparency, and color can show how strong and weak similarities between two vertexes is.

As for whether our method to calculate similarity is correct, it is questionable. There are parts where this method correctly calculates the similarities and there are parts where this method doesn’t get results higher enough so that two major that should have related are represented well. Also, please note that we use a high similarity cutoff to call that two majors are moderately or highly similar to each other.

However, since this paper's goal was to show the relationship/ similarity between undergraduate majors in ITB, this method works well to give a general sense of how these majors are related. Sadly, even though the author wants to find the most correct approach to calculate these similarities, that topic is outside the scope of this paper. Let us save it for another time.

VII. Appendix

The source code for this project could be found here: https://github.com/akbarmridho/majors-similarity.

VIII. Acknowledgment

This paper is possible thanks to all the lecturers of IF2120 Discrete Mathematics. K01 students, especially the author, would like to thank Dr. Nur Ulfa Maulidevi, S.T., M.Sc. as the designated class lecturer for K01. The author also would like to give thanks to the persons that develop the Spacy library and models, Circos’ author, the Nxviz author, and my laptop that make this paper possible to do.

References

[1] Krzywinski, M. *et al*, *Circos: An Information Aesthetic for Comparative Genomics*. Genome res (2009) 19:1639-1645.

[2] Munir. Rinaldi, *Graf (Bagian 1).* Bandung, West Java, 2022.

[3] IBM Cloud Education, *what is Natural Language Processing?* IBM (2022). Accessed on 6 December 2022.

[4] Bird. Steven, *et al*, *Natural Language Processing with Python – Analyzing Text with The Natural Language Toolkit*. O’Reilly (2009).

[5] Manning. D. Christopher, *et al*, *Introduction to Information Retrieval*. Cambridge University Press (2008).

[6] Mikolov, Tomas, *et al*, *Efficient Estimation of Word Representations in Vector Space*. arXiv (2013).

[7] Loukas. Serafeim, *Everything You Need to Know about Min-Max Normalization: A Python Tutorial*. Published on Medium (2020). Accessed on 6 December 2022.

[8] IAHPC Pallipedia. *Percentile*. IAHPC Pallipedia. Accessed on 6 December 2022.

[9] Munir. Rinaldi, *Graf (Bagian 2).* Bandung, West Java, 2022.

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, 9 Desember 2022

Akbar Maulana Ridho (13521093)