

Expanding Louvain Algorithm for Clustering Relationship Formation

Murniyati¹, Achmad Benny Mutiara², Setia Wirawan³, Tristyanti Yusnitasari⁴, Dyah Anggraini⁵
Computer Science and Information Technology of Gunadarma University, Jakarta, Indonesia^{1,2,3,4,5}

Abstract—Community detection is a method to determine and to discover the existence of cluster or group that share the same interest, hobbies, purposes, projects, lifestyles, location or profession. There are some example of community detection algorithms that have been developed, such as strongly connected components algorithm, weakly connected components, label propagation, triangle count and average clustering coefficient, spectral optimization, Newman and Louvain modularity algorithm. Louvain method is the most efficient algorithm to detect communities in large scale network. Expansion of the Louvain Algorithm is carried out by forming a community based on connections between nodes (users) which are developed by adding weights to nodes to form clusters or referred to as clustering relationships. The next step is to perform weighting based on user relationships using a weighting algorithm that is formed by considering user account activity, such as giving each other recommendation comments, or to decide whether the relationship between the followers and the following is exist or not. The results of this study are the best modularity created with a value of 0.879 and the cluster test is 0.776.

Keywords—Community detection; Louvain algorithm; modularity; network clustering relationship

I. INTRODUCTION

Social networks become a well-known instrument to disseminate information and to connect people who have the same thoughts. Public accessibility of this network with the capability to share opinions, thoughts, information and experience, offer tremendous promises for companies and governments. Apart from individuals who use this network to connect with friends or relatives, many companies and governments start to leverage social network platform to deliver their services to the society, citizen and client. Trust is a crucial issues of an effective social network. A social network is defined as a graph that contain set of nodes. Nodes represent objects, actors, people, or organizations while links express collaborations, communications or interactions. A complex network shows a very dense (such as a network of friendships or collaborator) or sparse network[1] [2][3][4].

Social network has received much attention and has been studied during the last decades including community detection in large and complex network[5] The aim of community detection is to divide the network and to illustrate this network in graph's form. The nodes (objects, actors, people, or organizations) that have relationship or correlations are said to be in the same community. Community can be used to obtain various purposes (such as finding targets who like similar product, encounter target markets, defining product ratings popularity, determining product recommendations and much

more [6]. There are lots of information on social media that can be used to cluster the data based on their similarity. Individuals can find peoples' biodata or what they share in their social media. This behavior definitely influenced the consequences of user curiosity. This research will form communities based on books at Gramedia Pustaka Utama.

Clustering is the assignment to separate the object inside the population into number of groups based on certain characteristic. Community will be detected based on the obtained information. Unfortunately, there are some drawback on the previous community detection algorithm. The previous algorithm could not perform well on large data set. Most studies, maximize the quality function to determine the community. This method known as modularity. In modularity, the nodes inside the same community are highly connected but loosely linked to nodes outside their community[7]. Modularity maximization calculates the quality of a particular clustering of a network into communities. It is a matter of processing speed and most of its algorithm use heuristic technique. Amongst the most efficient modularity maximization algorithm is using Louvain method. Louvain methods is a method to extract communities form large scale network. The results obtained from Louvain method give good modularity quality which describe the closeness relationship between nodes. This closeness relationship will be used to compute the confidence level of users in the recommendation system that will be developed[8][9]. Throughout developing Louvain algorithm, the usage of large amount data set from social media network is capable and will generate fast clustering relationship.

Social network influenced the behaviour of users. Therefore, in this research a recommendation system was developed based on the relations that were obtained from the social network[10]. The necessary step to implement clustering relationship formation was by performing scrapping. Scrapping was achieved to obtain relation on Gramedia Pustaka Utama account. The steps taken are scrapping first to get existing relationships with Instagram accounts, then doing weights and forming clusters by developing the Louvain algorithm. The purpose of this study is to develop the Louvain algorithm to produce clustering relationships so that you can see your circle of friends on Instagram social media. The results of this study can be used to produce a recommendation system that is not only based on ratings and reviews, but based on clustering relationships. Purity is used to measure the success of the clusters formed.

II. LITERATURE

A. Community Detection Algorithm

Detecting communities is a very essential task as communities helps us in grouping users showing similar behaviour and in this way the social network can be divided into different clusters of nodes with same behaviour. This community information can help us take useful decisions and extract important information about users in a particular community [11].

Community detection is an algorithm to create groups or partitions and to evaluate the formations of these groups. There are many community detection algorithm propose by different researchers, such as :

- Strongly connected components (SCC) are one of the initial graphing algorithms. This algorithm was described by Tarjan in 1972. SCC algorithm defined directed graph into strongly connected components which is a classic application of depth-first reach algorithm [12][13].
- Weakly Connected Components or Union Find algorithm found set of connected nodes in undirected graph, where each node is reachable from any other nodes in the same set. Weakly Connected Components is different from SCC. This algorithm only require an existing path between a pair of nodes in one direction, whereas SCC algorithm requires path to exist in both direction. Like SCC, Weakly Connected Component algorithm often used as initial stage in analyzing the graph's structure [13].
- Label Propagation Algorithm is a fast algorithm to discover communities in graphs. To detect a community using the structure of the network does not require any predefined objective functions nor prior information regarding the community. One of the interesting features of Label Propagation Algorithm is that the algorithm has the option to control the initial label to narrow down the obtained solution. At initial stage, each node has a unique label. Afterwards, labels are assigned iteratively to nodes in a random sequential order in such a way that nodes take the most frequent label of its neighborhood. The label relocation unites when there is no more alteration in the node label. Groups of nodes that have identical nodes at convergence form communities. Although this approach is efficient and does not require user-defined parameter, it is not deterministic due to the random choice of nodes to be labeled and the possibly large number of edges explored during the iterations [14].
- Triangle count and average clustering coefficient algorithm computes the number of triangles in the graph. Triangle is three set of nodes where each node is connected to the other two. In graph terminology triangle is known as 3-clique. The Triangle Count algorithm in Graph Data Science (GDS) library discover triangles only in undirected graphs. Triangle count has gained popularity in Social Network Analysis (SNA). This algorithm is used to detect communities and to measure

the cohesiveness of these communities. It also can be used to define the constancy of a network. Moreover, triangle count can be employed to compute the networks' indexes, such as clustering coefficient and local grouping coefficient [15].

- Another method for community detection is a spectral approach by Newman, which is a top-down hierarchical one that depends on eigenvectors of the modularity matrix. This approach works by iteratively separating the network into two components so that the modularity is maximized.[16]
- Another commonly used method for community detection is based on the modularity maximization , which calculates the quality of a particular clustering of a network into communities. The intuition behind modularity is that nodes inside the same community are highly connected but loosely linked to nodes outside their community.[17][18]
- Louvain modularity is an algorithm for detecting communities in a network. This algorithm maximizes modularity value of each community. Modularity quantifies the quality of assigning nodes to the community by evaluating how much more tightly connected the nodes in the community are, compared to how connected they would be in a random network. Louvain algorithm is one of the fastest modularity algorithm that perform well on large scale graph. This algorithm reveals the hierarchy of communities at different scales, which is useful for understanding the global functioning of the network. To understand Louvain's modularity algorithm, it is important to learn modularity in general.

It can be said that community detection is an effort and process to determine and find a group of people who have the same or the same interests. Community detection or clustering graph becomes part of data analysis in various fields; computer science, science, social network analysis and internet applications. As the data grows on exploratory power. Community detection is widely used in graph analysis. Given the graph $G = (V, E)$, the goal of the community detection problem is to identify the partitioning of nodes into "communities" (or "clusters") so that related nodes are assigned to the same community and different or unrelated nodes are assigned to the same community. Different. The community detection problem differs from the classic graph partition problem in that neither the number of communities nor their size distribution are known a priori. Due to its ability to uncover structurally coherent node modules, community detection has become a structure discovery tool in a number of scientific and industrial applications, including biological sciences, social networking, retail and finance.

The concept of community detection exists in network science as a method for finding communities in complex systems through graphical representations. The community detection method finds subnetworks statistically between nodes or graphs in the same community rather than nodes in different communities [19].

The core of community detection is the idea of modularity, a metric of which differs below:

$$Q = \frac{1}{2m} \sum_{i,j} A_{ij} \left(A_{ij} - \frac{k_i k_j}{2} \right) \delta(g_i, g_j) \quad (1)$$

In the above difference Q is modularity, A_{ij} is the edge weight between vertices i and j , k_i is the total weight of all edges connecting node i with all other vertices, and $2m$ is the total weight of all edges in the graph. The Kronecker delta function $\delta(g_i, g_j)$ will evaluate to one if the nodes i and j belong to the same group, and are zero. Modularity is the defining state of how one decides to divide the network. An unshared network is a network in which each node in its own community will have a modularity equal to zero. The goal of community detection is to find a community that can maximize modularity. There are many efficient algorithms to maximize modularity, including spectral clustering [16] [20]. Fig. 1 shows multiple networks with maximum increased modularity. Notice how the community structure becomes clearer as the value of modularity increases.

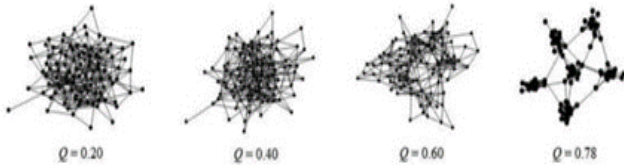


Fig. 1. Networks with different maximum modularity (Blondel et al., 2008)

B. Louvain Algorithm

Louvain's algorithm shows an algorithm that directly maximizes modularity with 2 phase algorithm. This first algorithm consists of nodes moving one by one in one of the neighboring communities to get the maximum increase in modularity, the nodes can be moved multiple times and this procedure stops if maximum locales are obtained, that is, when there is no more movement which increases the modularity. The second algorithm is the formation of a Meta graph where the nodes are the communities found in phase 1 and the links represent the number of connections between communities. The Louvain algorithm is an unsupervised algorithm that does not require input on the number of communities or size before running. The Louvain algorithm is divided into 2 phases, namely Optimizing Modularity and Community Aggregation.

Louvain's algorithm is one of many algorithms for community detection. One of the advantages of the Louvain Algorithm is that it detects communities with maximum modularity and is also faster than other algorithms.

Louvain's algorithm was first introduced to find the Newman-Girvan high partition modularity.

$$M = \frac{1}{2m} \sum_{i,j} (A_{ij} - p_{ij}) \delta(c_i, c_j) = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \quad (2)$$

A_{ij} is the neighbor matrix entry which represents the weight of the edge connecting vertices i and j , $k_i = \sum_j A_{ij}$ is the degree of the node i , c_i is the community, $\delta(c_i, c_j)$ value 1 if $c_i = c_j$ and 0 if otherwise. $m = \frac{1}{2} \sum_{i,j} A_{ij}$ is the sum of the weight of all sides on the graph.

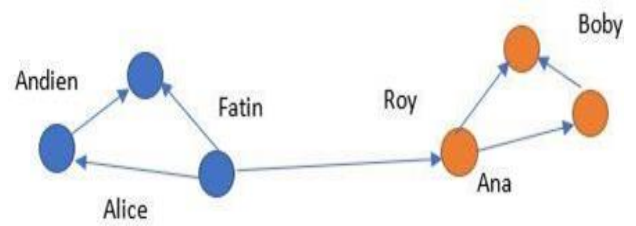


Fig. 2. Illustration of a community formed with louvain's algorithm

Louvain's algorithm in Fig. 2. finds two communities with three members on each community. Andien, Alice and Fatin are friends with each other, as are Roy, Ana and Bobby. Roy is the only one who has friends in both communities, but Roy has more friends who have the same characteristics in community two therefore Roy is in that community.

C. Modularity

Modularity is a measure of how well a group has been partitioned into clusters. It compares the relationships in a cluster against what is expected for a random number of connections. Criteria is known as modularity, its definition involves a comparison of the number of in-cluster links in a real network and the expected number of links in a random graph (regardless of community structure)[7].

III. LITERATURE

The steps taken to form clustering relationships are as shown in Fig. 3.

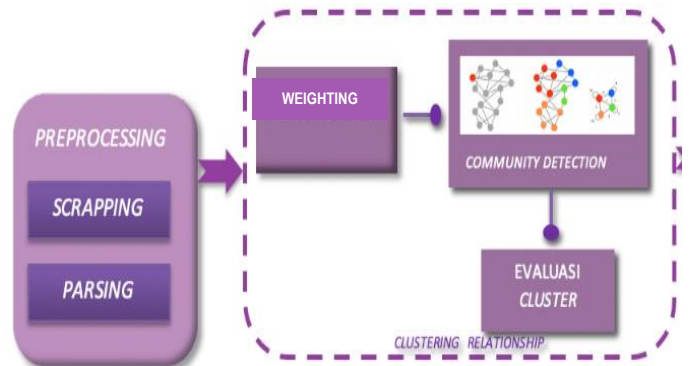


Fig. 3. System architecture

In Fig. 4, the system architecture begins with collecting data sets from Gramedia Pustaka Utama. The information obtained from Gramedia Pustaka Utama was related to the object of research, namely bookstores and novels.

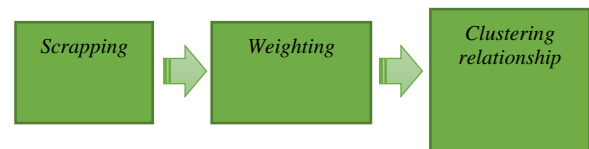


Fig. 4. Stages of forming a clustering relationship

After collecting the data set, the preprocessing stage is carried out using a social-based concept to take into account the relationship between users and other users in a social networking application service. The stage that is done is by

doing scrapping to get connectedness between users and aims to see how much users trust other users (shown in Fig. 5).



Fig. 5. Scrapping stage

Network connectivity on Gramedia Pustaka Utama was given a weight and then community detection is carried out using the Louvain method. The next step is to build a trust matrix which is represented by the results of weighting and relationship. This procedure is called a clustering relationship. The next stage is to compute the level of similarity of a user. In this stage every user will produce a rating prediction using collaborative filtering based similarity

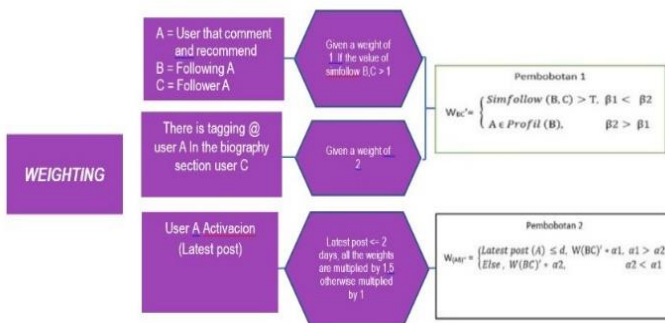


Fig. 6. Weighting process

The weighting procedure is a process that was performed to compute the trust value between users. The process was expected to obtain cluster's weight or value when creating a cluster relationship. This weight or relation value can be used in forming matrix trust. The weight is based on the relationship between A, B, C, users who recommend others and the activeness of user.

The weighting was performed in 2 stages. The first stage was computing the closeness of relationship using variable 1 as WBC (WBC weight). The given value for variable β_1 is 1 and for variable β_2 is 2. The second stage of the weighting process is to see user's activity. This step is require to obtain the users that are active to form a cluster. The variable that used for 2 is WAB (WAB weight). The given value for variable α_1 is 1.5 and for variable α_2 is 1 as shown in Fig. 6.

In algorithm, weight 1 is the weight for user A who has proximity (simfollow) between B and C > the threshold value, then it is being given weight of 1 and if in Biography C there is a tagging @user A then it is given weight of 2. The value of weigh 1 is set to variable β_1 , the weight value of 2 is set to variable β_2 where $\beta_1 < \beta_2$.

The next step is performed by increasing the weight level. For the second weighting, the weight can be given by looking at user activity (User A) from user A's last post. If user A's last post is less than 2 days, then all user A's weights are multiplied by 1.5, if not multiplied only by 1.

Algorithm 1: Weighting Algorithm

Input : User A

Output: Followers A(C)
Following A (B)

1. Get Follower A
2. Get Following B
3. Give a weight of 1, if the followers intersect with the following, and give a weight of 2 if the Follower C user has a user A tag in the biography section, using a weighting formula 1:

$$W_{BC} = \begin{cases} Simfollow(B,C) > T, \beta_1 < \beta_2 \\ A \in Profil(B), \beta_1 > \beta_2 \end{cases}$$

4. If the latest post is less than or equal to 2, the weight is multiplied by 1.5 for users connected to the A besides that the weight is multiplied by 1, using the weighting formula 2 :

$$W_{AB} = \begin{cases} Latest Post(A) \leq d, W(BC)' * \alpha_1, \alpha_1 > \alpha_2 \\ Else, W(BC)' * \alpha_2, \alpha_2 < \alpha_1 \end{cases}$$

The multiplier of 1.5 is assigned a variable (α_1) and the multiplier of 1 is assigned a variable (α_2). The value of $\alpha_1 < \alpha_2$. This function is to indicate the activeness of the user. For example, user Osterdamm (A) comments on the book 'The Miracle of Mindbody Medicine_New', (B) is a user in Following Osterdamm, (C) is all Follower Osterdamm users if they are related or are friends then they are given a weight of 1, if Follower Osterdamm in his biography has the tag '@Osterdamm' then it is given a weight of 2, if Follower Osterdamm last posted less than 2 days then the weight is multiplied by 1.5.

IV. RESULT

Community detection is a method to find communities in a large and complex network. This method optimized modularity. There are many algorithm to maximize modularity. For example spectral clustering (Newman, 2006) and fast unfolding (Blondel et al., 2008). This research propose Louvain algorithm to detect communities. Louvain algorithm is considered the most suitable method to detect the communities since the algorithm works well on large and complex network. This algorithm does not require data input (Unsupervised learning). Moreover the algorithm can form clusters/communities faster compare to other algorithms. Louvain algorithm is the development of the existing community detection algorithms. Louvain is unsupervised learning algorithm. It does not require input of communities' size or number. The algorithm is divided into two phase; Modularity optimization and community aggregation. Modularity is used in this research to measure how well a group has partitioned into clusters.

Below is the equation to compute modularity:

$$M = \frac{1}{2m} \sum_{i,j} (A_{ij} - p_{ij}) \delta(c_i, c_j) = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \quad (3)$$

A_{ij} is the adjacency matrix that represent the edges weight which connect node i and node j , $k_i = \sum_{j} A_{ij}$. A_{ij} is the degree of node i , c_i is its community, δ -function $\delta(u, v) = 1$ if $u = v$ and 0 otherwise. $m = \frac{1}{2} \sum_{i,j} A_{ij}$ is the sum of all weight in a graph.

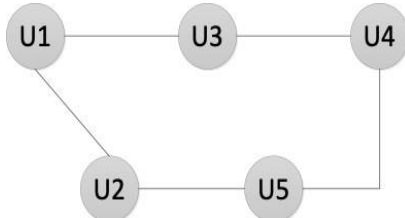


Fig. 7. A simple graph

Fig. 6 shows a graph with 5 nodes (U1, U2, U3, U4 and U5) and 5 edges (U1-U3, U3-U4, U4-U5, U5-U2, U2-U1). According to the definition of Neighbor Matrix explained by A_{ij} in the previous section, it will look like Table I.

TABLE I. TABLE GRAPH REPRESENTATION IN MATRIX

	U1	U2	U3	U4	U5
U1	0	1	1	0	0
U2	1	0	0	0	1
U3	1	0	0	1	0
U4	0	0	1	0	1
U5	0	1	0	1	0

Considering assigning the partition as:

Partition 1, U1-U1, U3, U4

Partition 2, U2-U2, U5

Using Modularity equation (1)

1) Node U1 to U1

$$Q = 1/2 * 5((0 - 2 * 2)/(2 * 5)) * 1$$

#node U 1 – U 1

Same member (one partition)

2) Node U1 to U2

$$Q = 1/2 * 5((1 - 2 * 2)/(2 * 5)) * 0$$

#node U 1 – U 2

Not the same member (not one partition)

3) Node U1 to U4

$$Q = 1/2 * 5((0 - 2 * 2)/(2 * 5)) * 1$$

#node U 1 – U 4

Same member (one partition)

4) Node U1 to U5

$$Q = 1/2 * 5((0 - 2 * 2)/(2 * 5)) * 0 \text{ #node U 1 – U 5}$$

Not the same member (not one partition)

5) Node U1 to U3

$$Q = 1/2 * 5((1 - 2 * 2)/(2 * 5)) * 1$$

#node U 1 – U 3

Same member (one partition)

Modularity optimization in this research is illustrated as follows. \sum_{in} is the sum of edges' weight in C , \sum_{tot} is the sum of all edges to node in C , k_i is the sum of weight from all edges in node i , k_i, in is the sum of edges from node i to node inside community C and m is the sum of weights from all edges in the graph.

Louvain Algorithm PASS 1 and Louvain Algorithm PASS 2. The first step in Louvain Pass 1 algorithm is to select the initial node and computing the modularity transformation that might occur when a node join and formed community with this node close neighbor. The next step is that initial node join the node with the highest modularity change. This process will be repeated for each node until the community is formed. Communities are combined to create super communities and relationships between this super node counts as the sum of the previous links (Self-loop represents the previous relationship and is now hidden in the super node) The resulting graph using the Louvain PASS 1 algorithm can be seen in Fig. 7.

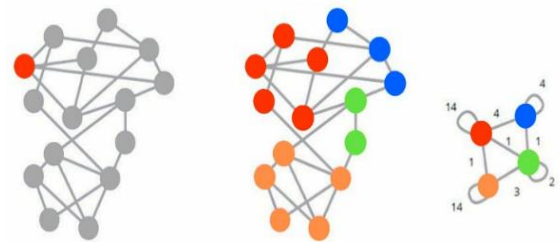


Fig. 8. The created graph with PASS 1 algorithm

The steps for creating a graph in the Louvain PASS 1 algorithm are as follows:

Algorithm 2: LOUVAIN PASS 1

Require : $G = (V, E, w)$ a weighted graph

Ensure : a partition P of V

Local : increase \leftarrow true

Local : P current partition of V

Begin

Forall the nodes I do

$P[i] \leftarrow \{i\}$

INIT (i)

While increase do

increase \leftarrow false

forall the nodes i do

Remove (i, C_{old})

$C \leftarrow P[j] \mid (i, j \in E) \cup \{C_{old}\}$

$C_{old} \leftarrow P[i]$

$C_{new} \leftarrow \operatorname{argmax}_{c \in c} \{GAIN(i, C)\}$

INSERT (I, C_{new})

if $C_{old} \neq C_{new}$ then

increase \leftarrow true

The resulting graph using the Louvain PASS 2 algorithm can be seen in Fig. 8 below:

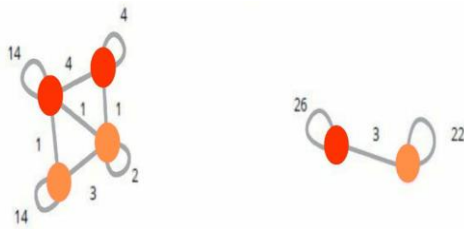


Fig. 9. The created graph with PASS 2 algorithm

The steps for creating a graph in the Louvain PASS 2 algorithm are as follows:

Algorithm 3: LOUVAIN PASS 2

Require : $G = (V,E,w)$ a weighted graph

Ensure : a partition P of V

Begin

Repeat

$P \leftarrow \text{ONEPASS}(G)$

$G \leftarrow \text{Partition} - \text{to} - \text{Graph}(P,G)$

until no improvement is possible

In the Louvain PASS 2 algorithm, steps 1 and 2 are repeated in the path until no further growth in modularity. The PASS 2 algorithm will also repeated until the number of iterations has occurred.

In the initial research conducted using the Louvain PASS 1 and PASS 2 algorithms, the resulting graph can be seen in Fig. 9 and Fig. 10.

Modularity for best partition: 0.9568860789869055

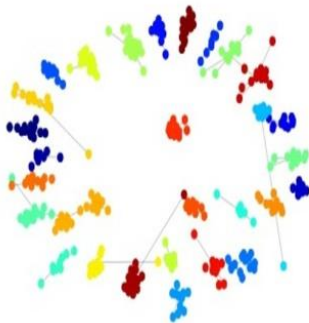


Fig. 10. Visualization of clustering relationship using the louvain pass 1 algorithm

Fig. 9 shows the obtained graph using Louvain algorithm. This figure created from some of the data (219570 data from Gramedia Pustaka Utama’s users). The visualization of Louvain PASS 2 algorithm can be seen in Fig. 10. The obtained value of modularity is 0.5, indicating that the community detection was not good enough.

The proposed Louvain algorithm. In the previous Louvain Algorithm, there are 2 phases of modularity formation. The developed algorithm starts from the input which is the result of weighting the nodes to form communities. The formation starting by defining different communities of each network node. In the initial partition, the number of communities is equal to the number of nodes.

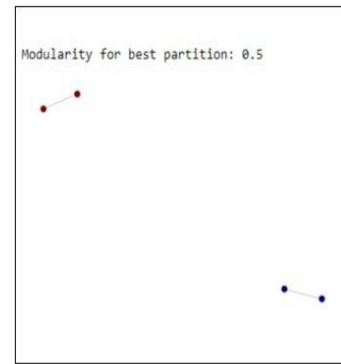


Fig. 11. Visualization of clustering relationship using the louvain pass 2 algorithm

Algorithm 4: LOUVAIN MURNI

Input : Dataset of relations between Instagram users

Output : Clustering Relation Graph, list of cluster member

1. Random user to be used as a start point (make S variable)
 2. Remove users who are neighbors with S (save to variable N) from cluster 1 to a new cluster (cluster 2)
 3. Compute modularity (make it mod_new variable); using the modularity formula:
- $$M = \frac{1}{2m} \sum_{i,j} (A_{ij} - p_{ij}) \delta(c_i, c_j) = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j)$$
4. If mod_new is greater than mod , make $X=X+1$ and set $mod = mod_new$, if smaller than return to cluster 1
 5. Set $S = N$
 6. Repeat step (4) ,until all users are counted

For each node i , the neighboring nodes i , namely node j will be considered. Then the value of obtained modularity will be evaluated by removing node i from its community and then placing node i in the community of node j . Node i , then positioned in the community that provides the greatest value of profit, but only if the value of the gain is positive. If there is no possibility of a positive profit value, node i will remain in the community from which the node originated. This process will be repeated successively for all nodes until no further improvement can be achieved. With the fulfillment of these conditions, the first phase of the algorithm in this study has been completed.

The previous Louvain algorithm which consists of 2 Louvain phases produces a super cluster which is only a few large clusters. Pure Louvain Algorithm which was developed to get user clusters that have relationships obtained from the previous weighting process, this is needed for the stage of forming a recommendation system based on Trusted Friend.

The results of the pure Louvain Algorithm will be represented in the form of a graph, a list of cluster members and the relationships between users

V. PURITY APPROACH

Purity or a measure of purity historically was the first measurement used in the context of community detection used by Girvan and Newman in their article. Purity has gone by a variety of different names in several articles making it difficult to name a complete list. The purity of a part relative to part Y is expressed in the following equation:

$$Pur(x, Y) = \max \frac{nij}{ni+} \quad (4)$$

In other words, the first thing to do is to identify the part with the largest intersection and then calculate the proportion of the elements. The greater the intersection and the greater the purity value, the greater the correspondence between the two parts being analyzed. Then the total partition X relative to the partition is obtained by adding up the purity of each xi, then given a weight using the following equation:

$$Pur(X, Y) = \sum_i \frac{ni+}{n} Pur(xi, Y) \quad (5)$$

The upper limit on purity is 1, which corresponds to a perfect match between each partition, while the lower limit is 0 which is the opposite value of the upper limit. Purity is not a symmetrical measure, meaning that in the process, purity is relative to the amount considered in each part. Therefore, in general PUR(X,Y) is not the same as PUR(Y,X).

From a community detection point of view, two different purity measurements can be used, depending on whether to calculate the estimated community purity relative to the true value, or vice versa. In cluster analysis, the first version is generally used, and is called simply Purity, while the second version is Inverse Purity. It is difficult to determine which one is actually used in the case of existing community detection. Girvan and Newman provide a very concise description of the size being processed. Purity tends to favor algorithms that identify many small communities. In the most extreme case, if the algorithm identifies n communities containing one node each, one of the clusters gets the maximum purity, because each estimated community is perfectly pure. In contrast, reverse purity supports algorithms that detect multiple large communities. The most extreme case occurs when the algorithm places all nodes in the same community, then a cluster gets the maximum purity, because each community is actually perfectly pure: all the nodes it belongs to belong to the same estimated community. To solve this problem, Newman introduces an additional solution: when the estimated community is majority in some actual community, all the nodes in question are considered to be the wrong classification. The solution generally adopted in cluster analysis consists mainly of processing the F-Measure, which is the average of the harmonics of the two purity versions:

$$F(X, Y) = \frac{2.Pur(X*Y*Pur(Y,X))}{Pur(X,Y)+Pur(Y,X)} \quad (6)$$

The measure obtained from the above equation is symmetrical, and this combination is expected to resolve the aforementioned bias. This approach provides a solution in a similar way by underestimating and overestimating the number of communities. The purity value is calculated for all clusters formed, namely 27 clusters. Calculating the purity value,

starting from the cluster that has a majority value to users who tag the preferred book, as shown in Table II below.

TABLE II. THE OVERALL PURITY OF THE CLUSTER

	Lower Bound	Maximum	Purity
Cluster 1 until Cluster 27	0.0	1	0.776

For example, Cluster 1 the majority value is 25, Each cluster is calculated as the majority value, so that the total of the entire cluster has a purity value of 0.776 as shown in Fig. 11:

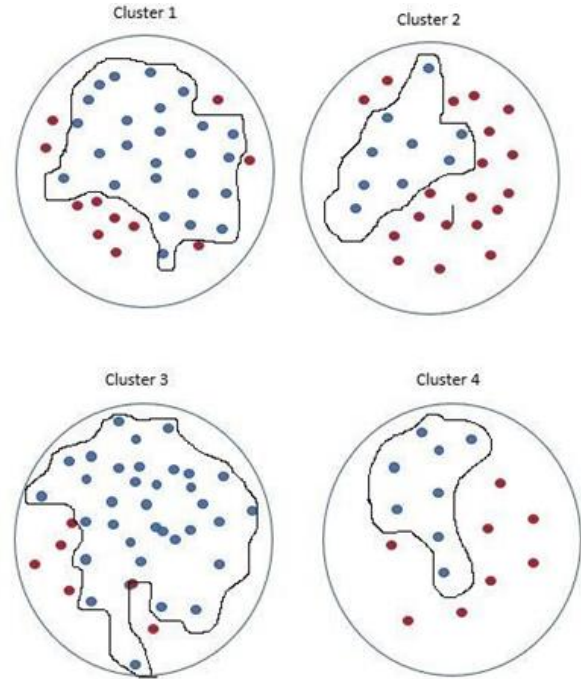


Fig. 12. Purity as an external evaluation criterion for the quality of cluster 1 to cluster 27.

VI. CONCLUSION

Based on a number of tests and analysis of the results of this study, it can be concluded that:

- 1) The social network on Gramedia Pustaka Utama account forms a clustering relationship that is used for the recommendation system by weighting and community detection. The weighted value (α,) given is proven to affect the results of the community that is formed.
- 2) Clustering Relationship succeeded in forming clusters using the Louvain algorithm, as many as 27 clusters with the best value of high modularity, namely 0.879.
- 3) The value of the modularity of the community that is formed is influenced by the number of relationships between community members where the denser the relationships in the community, the value of modularity will increase or be higher.
- 4) The use of algorithms with modularity optimization has slightly better results because modularity shows how well the community on the network is.

5) Evaluation of the cluster formed using purity produces a satisfactory value of 0.776.

ACKNOWLEDGMENT

The authors would like to acknowledge to Research Center of Gunadarma University and Gramedia Pustaka Utama. The authors also thank to colleagues for input and support.

REFERENCES

- [1] W. Pedrycz and S.-M. Chen, Eds., *Social Networks: A Framework of Computational Intelligence*, vol. 526. Cham: Springer International Publishing, 2014.
- [2] Murniyati, A. B. Mutiara, S. Wirawan, and T. Yusnitasari, "Social Networks for Establishing Clustering Relationships on Instagram Social Media," *I T A L I E N I S C H*, vol. 11, no. 2, pp. 394–407, 2021, doi: 10.1115/italienisch.v11i2.132.
- [3] S. Lobanova and A. Chepovskiy, "Combined method to detect communities in graphs of interacting objects," no. 4 (42), pp. 64–73, 2017. [Online]. Available: <https://bijournal.hse.ru/en/2017--4>.
- [4] L. Ma and Y. Zhang, "Hierarchical social network analysis using multi-agent systems: A school system case," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 1412–1419, doi: 10.1109/SMC.2014.6974113.
- [5] Y. W. and Q. Y. Y. Zhang, "Community Discovery in Twitter Based on User Interests," *J. Comput. Inf. Syst.*, vol. 8 N0. 3, pp. 991–1000, 2012.
- [6] W. M. and A. A. C. N. Utami, "Analisis dan Implementasi Community Detection Menggunakan Algoritma Girvan and Newman Dalam Sosial Network," 2013.
- [7] Newman, M.E.J., "Modularity and community structure in networks," *P. Natl. Acad. Sci. USA*, vol. 103, pp. 8577–8582, 2008.
- [8] A. Stachowiak, "Propagating and Aggregating Trust with Uncertainty Measure," 2011.
- [9] A. Stachowiak, "Uncertainty-Preserving Trust Prediction in Social Networks," 2014.
- [10] F. E. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," *Auton. Agent. Multi. Agent. Syst.*, vol. 16, no. 1, pp. 57–74, 2008, doi: 10.1007/s10458-007-9021-x.
- [11] P. Kumar, R. Jain, S. Chaudhary, and S. Kumar, "Solving Community Detection in Social Networks: A comprehensive study," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 239–345, doi: 10.1109/ICCMC51019.2021.9418412.
- [12] S. Raghavan, "Twinless Strongly Connected Components," in *Perspectives in Operations Research: Papers in Honor of Saul Gass' 80th Birthday*, F. B. Alt, M. C. Fu, and B. L. Golden, Eds. Boston, MA: Springer US, 2006, pp. 285–304.
- [13] I. T. Neo4J, "Neo4j Graph Data Science Library," 2022. <https://neo4j.com/docs/graph-data-science/current/algorithms/louvain/>.
- [14] B. Laassem, A. Idarrou, L. Boujlaleb, and M. Iggane, "Label propagation algorithm for community detection based on Coulomb's law," *Phys. A Stat. Mech. its Appl.*, vol. 593, p. 126881, 2022, doi: <https://doi.org/10.1016/j.physa.2022.126881>.
- [15] S. Putrevu, "A Note on Detection of Communities in Social Networks," 2020, doi: 10.18535/ijecs/v9i03.4449.
- [16] B. A.-L. and W. D. J. Newman M E J, "The Structure and Dynamics of Networks," *Princet. Univ. Press. Princet.*, 2006.
- [17] M. Newman, MEJ and Girvan, "Finding and evaluating community structure in networks," 2004, doi: 10.1103/Phys-RevE.69.026113.
- [18] J. Yang, J. J. McAuley, and J. Leskovec, "Community Detection in Networks with Node Attributes," *CoRR*, vol. abs/1401.7267, 2014, [Online]. Available: <http://arxiv.org/abs/1401.7267>.
- [19] S. Zhang, Z. Liu, and W. Dou, "An Enhanced Community Detection Method Based on Neighborhood Similarity," in *Proceedings of the 2012 Second International Conference on Cloud and Green Computing*, 2012, pp. 493–500, doi: 10.1109/CGC.2012.71.
- [20] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, p. P10008, Oct. 2008, doi: 10.1088/1742-5468/2008/10/P10008.