The kindness behavior management in kindness service application using tree structure

Luxfy Roya Azmi, Lukito Edi Nugroho, Adisty Erna Permatasari
Dept. Electrical Engineering and Information Technology, Universitas Gadjah Mada, Yogyakarta, Indonesia

*luxfy.roya.a@mail.ugm.ac.id

Abstract. Kindness service application is an application which helps the students in kindness behavior implementations. This application is designed as an intermediary between kindness service providers and those who need kindness service, so it constructs a two-way relationship between the students. In addition, this application can also motivate the students to compete to do good things, as those who often doing kindness will be given a rate by the admin. The process of kindness behavior that had occurred will be designed to form a kindness tree. The management of this process is designed by using tree structure method. The tree structure will set the formation of kindness behavior in the kindness service application. Therefore, kindness behavior that had occurred will form into a whole kindness tree even though it provides the features to add or remove the behaviors of kindness.

1. Introduction
The development of technology in the digital era provides easy access to internet via mobile phones. Based on the statistics of mobile phone usage in Indonesia, more than 74.9 million people [1] is using mobile phone. It causes a negative impact on their social interactions among individuals and the change of values in social life, particularly among the students and adolescent. The interaction among the students has decreased and social contact also becomes lower. One of the shifting patterns of relationship among students and the surrounding environment is the reduction in helping behavior in social life. Therefore, an innovation is needed to be applied as representation of a culture of mutual help that is in line with the development of digital technology.

The alignment between technological developments with the implementation of kindness behavior, a digital innovation is needed to connect students who need kindness behavior with students who give kindness behavior that makes a culture of mutual help which can be digitally helped. To implement digital interaction to run properly, it is necessary to design a kindness behavior application which later becomes the main media in the implementation of digital kindness behavior.

The Implementation of kindness behavior can occur by connecting two students who need each other kindness so that a two-way relationship between students is formed. In addition, the system can motivate students to compete to do kindness because students who often do kindness will be given a rate by the admin.

The design of kindness behavior system consists of three components. Those are the design of admin functions, architectural design and management of kindness behavior. The component of admin function includes admin management on the system and designing kindness behavior by automatically verify the kindness behavior that enters the system and presents kindness behavior that has occurred
by giving a rate in accordance with the behavior of the kindness that has been done. Architectural design component in the management of the distribution of data and information on the system makes the component of management of kindnessness will work properly and presents the kindness that is mutually bound. Kindness behavior management is the core of this research.

Management of kindness behavior that has occurred in the system will be arranged to form a kindness tree scheme using a binary search tree algorithm. Binary search tree will determine the arrangement of kindness behavior started from the main kindness behavior to the behavior of the supporters of the kindness. The kindness that happened will be rated by the admin in accordance with the level of kindness behavior.

Implementation of binary search tree equations is useful for efficient hierarchical data manipulation and developing concurrent algorithms for derived data structures such as interval trees on kindness behavior. Concurrent binary search tree (CBST) is a searching tree that supports concurrent execution of all basic operations, such as insert, delete, and search, as mentioned in [2] [3] [4] [5] [6].

The use of binary tree is necessary to manage kindness behavior, in the application of kindness. Binary Tree function is to form a kindness behavior so that the kindness becomes structured from the main kindness to the kindness of the supporters in the application of kindness services. To determine the structure of kindness behavior that is new to the kindness service application, every kindness is added to the system that will be filtered by the admin, so the kindness behavior that enters the system will look for relationships with kindness behavior on the system. The kindness behavior that is added to the system has no correlation to the system. Therefore, the behavior of kindness is single system.

2. Literature Review

To get more specific information about tree structure, it is necessary to understand previous studies that are relevant to the research that will be conducted, Drachsler et al [3], propose a concurrent binary search tree based on high balance keys. Each node in a CBST is like maintaining a pointer to its predecessor and its replacement node along with a pointer to parents and children in the tree. CBST was proposed by Crain et al. [4] is a friendly tree of contention. This is not a high balanced CBST and the implementation runs continuously explicit threads. This utility physically removes logically deleted nodes and reduces the difference between the height of the left sub-tree and the right sub-tree on each non-leaf node by using a recursive first-depth procedure.

CBST search operation suggested by Bronson et al. [6] requires hand-over-hand optimistic validation, which negatively impacts performance with increasing tree size. CBST proposed by Ellen et al. [5] is an external binary search tree that does not block. It doesn't rotate anything to maintain high balance. When the size of a tree is small, external trees have an advantage over internal trees because updates are only carried out at the leaf level or one level above. This reduces the dispute about trees. But for large trees, this results in a high average access point length in successful searches, which causes a decrease in performance.

Jim Myong Kim et al. [7] graph tree is used to determine graph isomorphism, all possible sequences of graphs are presented as all tree paths, all sequences that occur in the graph are examined by evaluation functions that determine effectiveness sub graph. Based on the literature review of the use of tree structure, in this paper the tree structure is used for kindness behavior management which is used to determine the kindness binary tree that can present the relationship of kindness behavior that is found in the kindness behavior system. Binary trees can search, add and delete on systems of kindness behavior so that the system keeps running well.

3. Components Of Kindness And How It Comes

A. The concept of two-way relationship

The concept of two-way relationship is a concept that is built to provide accurate services in accordance with what is needed by the user. The process of two-way relationship in the implementation system behavior can be seen in Figure 1.
Figure 1. Two-way relationship in kindness behaviour system

The stages of using the concept of two-way relationship in the system of kindness implementations begin in the process of entering requests for kindness behavior in the form of offers and requests for help. Then the system spreads the request for kindness behavior so that it finds a receiver that matches to those who request the behavior of the kindness. After a suitable receiver is found, then the details of the data from the receiver such as name, telephone number and type of kindness behavior that will be used arranged to be given to the user who requests kindness as the output of the system.

B. Components of Single Kindness

Kindness behavior consists of a single kindness behavior and the behavior of a group's kindness. Each kindness behavior can be a direct interaction of kindness if it consists of kindness behavior and supporting attributes.

Figure 2. Kindness behaviour components

From Figure 2, kindness behavior has the attribute to carry out the kindness behavior. The user of kindness must fill his attributes according to the behavior needs of kindness, and after the user attribute matches, there will be a process of kindness and after the kindness happens it will be saved directly to the database without filling Id father and Id child because the kindness is single.

C. Components of the Group of Kindness Behavior

Behavior of group kindness is a kindness behavior in which more than one kindness behavior involved to be implemented. The kindness of this group is arranged based on filtering from the admin in the order of kindness behavior based on the level of kindness behavior which is arranged based on tree structure.
From Figure 3, it shows the behavior of group kindness consists of several kindness behaviors to choose the kindness of the group, so it must do the process of kindness as a whole in the tree structure above which consists of the main kindness to the kindness of the supporters. The kindness behavior that has occurred will be directly stored in the system database.

**D. The Process of Kindness**

To do the process of the occurrence of kindness behavior, there are several steps that must be done in implementing kindness behavior on the system so that those who need kindness behavior with those who provide kindness behavior can interact in implementing kindness behavior.
Figure 4. The Process of Kindness

Figure 4, figures out how the process of kindness occurs in the system. From the picture there are two processes of kindness behavior, namely offering help and asking for help. In the process of offering the kindness behavior, the user fills in the required attributes on the blackboard to find user who want to accept the kindness behavior, while in the process of asking for help the user sends what is needed to the blackboard and the recipient of kindness behavior immediately fills the attribute to do the kindness process.

4. Methods

A. Binary Search Tree

A binary search tree is a binary tree that has an additional property consisting of elements in the left sub-tree of the right sub-tree [2]. In the service application kindness every process of occurrence of kindness will be stored in the database both single kindness behavior and group kindness behavior,

Figure 5. Binary search tree

In Figure 5 the kindness tree that has been formed in the system, there are three important operations on kindness behavior. These are search operations, adding new kindness behavior, and removing elements of kindness behavior.
1. Search operation

To determine the searching operation of binary search tree on kindness behavior data, we first compare the key data to be searched with the key from the root data if it does not match to the list of kindness trees. Then it looks for the left and right subpages until the key data you want to find is match. [3] So, every kindness behavior can find its part on the list of kindness behaviors that exist in the tree of kindness by using the Id of kindness behavior on the system and the id which is not suitable in the kindness tree , belongs to the single kindness and directly stored in the database.

2. The operation of adding elements to the kindness behavior.

The addition to the binary search tree in the first Id of kindness behavior, should be compared to the key from the root of the tree in question [3]. In the kindness behavior tree, the kindness of addition can only be done in the sub-tree in the bottom level of image 4.1 and it can be seen the Id behavior of kindness that can be added to the Id (4, 5, 6,7).

3. Element removal operation

Taging down or removing the elements in the binary tree is the most complex operation of these three operations. In this element of kindness behavior, we limit the subpages-deleted to the kindness tree. The kindness Ids that can be deleted are only the lowest level of kindness behavior Id (4,5,6,7) and Id that has a child of kindness cannot be removed because it can damage the function of the other kindness behavior.

B. Adjacency List

Adjacency list is used to determine each relationship on the kindness that happens, that can detect which kindness interrelated. It also can be given a sign Id of father and Id of child [8]. The Id of child will look for Id of father on the system if the new kindness has no correlation with the kindness found in the system of kindness will stand alone or a single kindness.

![Figure 6. Kindness Id Relationship](image)

To describe the relationship of each Id kindness starting from Id father to Id child kindness, it can be presented in the following table.

| Id Father | Adjacency | Id Children |
|-----------|-----------|-------------|
| 1         | Adjacent to | 2,3         |
| 2         | Adjacent to | 4,5         |
| 3         | Adjacent to | 6,7         |
From table 4.1, we can see the father (1) has children (2,3), and (2) has children (4,5) and (3) has children (6,7). The adjacency child who will look for the Id of each child from kindness as in the table above. If the kindness chosen is the kindest or the best of the child, the Id of the child will look for the father's Id, for example the child's ID (6) then the father's Id (3) and so on which can be seen from Figure 6.

5. Conclusion
In this paper, it explains how the process of kindness occurs that can connect students who provide kindness services with those who need the kindness, so that two-way relationship is formed between students and management of kindness behavior that has taken place in the form of a kindness tree which is stored in the database. The formation of the kindness tree is arranged using a tree structure where the method can form kindness behavior that occurs automatically on the system so that the system continues the function properly even though there are some elements added and deleted.

References
[1] Badan pusat statistik (2016, desember) “Pengguna Handphone di Indonesia”, from http://www.bps.go.id
[2] A. Natarajan and N. Mittal, “Fast concurrent lock-free binary search trees,” PPoPP, pp. 317–328, 2014.
[3] D. Drachsler, M. Vechev, and E. Yahav, “Practical concurrent binary search trees via logical ordering,” pp. 343–356, 2014.
[4] T. Crain, V. Gramoli, and M. Raynal, “A contention-friendly binary search tree,” in ICPP, pp. 229–240, 2013.
[5] F. Ellen, P. Fatourou, E. Ruppert, and F. van Breugel, “Non-blocking binary search trees,” PODC, pp. 131–140, 2018.
[6] N. G. Bronson, J. Casper, H. Chafi, and K. Olukotun, “A practical concurrent binary search tree,” pp. 257–268, 2010.
[7] Jim Myoung Kim dkk, “Tree search based determination of graph isomorphism” international conference on future generation communication and Networking, pp: 53-55, 2014
[8] Diestel, Reinhard. 2000. Graph Theory. New York: Springer-Verlag