

COMPUTER AIDED LEARNING FOR DATA STRUCTURE PROGRAMMING SUBJECT

Yulia¹, Rudy Adipranata² and Herman³

^{1,2,3}Universitas Kristen Petra, Indonesia, ¹yulia@petra.ac.id, ²rudya@petra.ac.id

ABSTRACT. Data Structure is one of the compulsory subjects in Informatics Department Petra Christian University (PCU). The students have a difficulty to understand Data Structure material if they just reads from the slides. Their difficulty is because the working process of the system is not visible, so the students must imagine it themselves. In this paper, we offer a Computer Aided Learning application made in the form of animated Data Structure modules, so the students can understand the material better. Besides animation, this application also displays the explanation material in each line of the codes. This application is built using Microsoft Visual Basic. Survey shows that 75% of the students understand Data Structure modules better after trying this application.

Keywords: Computer Aided Learning, Data Structure, Tools

INTRODUCTION

Data structure is one of the compulsory subjects in the department of Informatics Engineering PCU. In this course students learn to understand the various forms of data structures and algorithms and its implementation, in order to implement these algorithms and data structures to solve programming problems. This course became a prerequisite of many other courses in the Informatics Department. Learning Data Structure is very difficult for many students. Based on data survey to the student, there are approximately 30% of students who repeat Data Structures classes, and also 50% of students surveyed received only a C + or C grade for their Data Structures course.

The ability of each person in capturing the learning materials is not the same. There are people who can capture the subject matter quickly and some are slow. Thus the visual system will be created whose purpose is to facilitate students learn Data Structure.

In the current learning system of learning, the lecturers have to draw on the board for the students more clearly the workings of a data structure. In the new system, expected the lecturers do not need to draw on the board, just need to show animations using LCD projectors.

Therefore, it necessary to develop a CAL application that can help students understands the material of Data Structure in the form of animation that is more attractive to become a means of learning.

PREVIOUS WORK

Figure 1 below is an example of tools that support learning for subject Design and Analysis Algorithm, (Budhi, 2008). On the left is the source code of the DFS Iterative while the image on the right is example of nodes. When the program starts, it will show the tracing process of source code. While the image on the right will move according to the code that was highlighted on the left. Code that was in the left part can not be changed, because it is the source code of the Iterative DFS, while the image on the right can be arranged

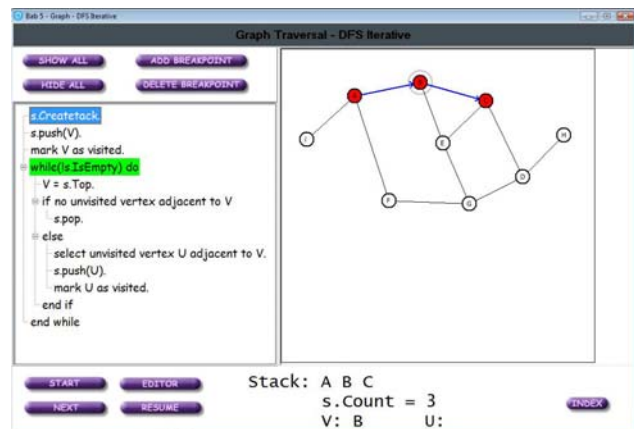


Figure 1. DFS Interactive Tools
(source: Budhi, 2008)

Figure 2 is example of tools that illustrate the making of the stack.

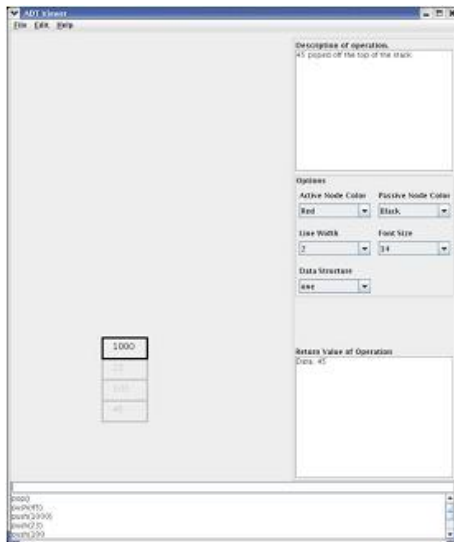


Figure 2. Tools for Making a Stack
Source: Lehman, 2005

THEORY

Stack

A stack is a data structure in which all access is restricted to the most recently inserted element. It behaves much like a stack of bills, stack of plates, or stack of newspapers. The last item added to the stack is placed on the top and is easily accessible, whereas items that have been in the stack for while are more difficult to access. Thus the stack is appropriate if we expect to access only the top item; all other items are inaccessible. In a stack the three natural operations of insert, remove, and find are renamed push, pop, and top (Weiss, 2003).

Queue

Another simple data structure is the queue, which restricts access to the least recently inserted item. In many cases being able to find and/or remove the most recently inserted item is important. The basic operations supported by queues are enqueue, or insertion at the back of the line, dequeue, or removal of the item from the front of the line; and getFront, or access of the item at the front of the line (Weiss, 2003).

Recursive

A recursive function is a function that either directly or indirectly makes a function call to itself. This action may seem to be circular logic: How can a function F solve a problem by calling itself? The key is that the function F calls itself on a different, generally simpler, instance (Weiss, 2003).

Binary Tree

A binary tree is a tree in which no node can have more than two children. Because there are only two children, we can name them left and right (Weiss, 1996). Figure 3 illustrates an example of a Binary Tree.

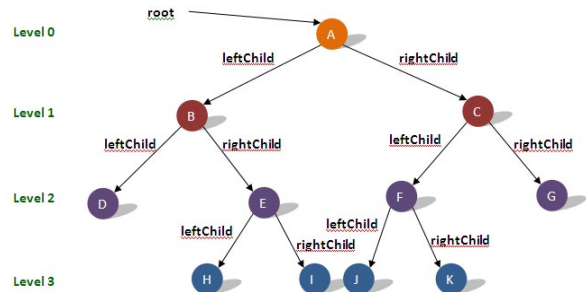


Figure 3. A Binary Tree

Hashing

Hashing is used by programmers when store and retrieve large amounts of data efficiently. Hashing is an array consisting of pointers that point to a data. While the data itself takes the form of a structure that is defined by the user itself, and consists of a maximum of three elements, namely key, value, and pointer to next structure in a hash. The pointer is only used when there is a collision in a hash. Collision occurs when the hash value generated by two or more keys that have the same value. The common method used is to create a linked list to two or more keys that have the same hash value.

Linked List

Linked List is used when going to save the data with the number who have not known exactly and can be added continuously. These data should be interconnected and have the same type. Linked List needs to be dynamically allocated and linked from the first data into second data, and so on.

COMPUTER AIDED LEARNING

Some advantages of using computers as a tool in the learning process are (Adipranata, 2006):

1. Individual Learning. Every person has a different way to learn. Every person also has a different ability to understand something. For people who need a long time in learning, will require time for learning independently. With the CAL, can help them to learn independently.

2. High public interest in using computers. People have a high interest in using computers. If the computer is equipped with exciting teaching applications, is expected to attract students in learning.

3. Learning fast. Privilege the use of computers in the learning process is the time required to learn anything faster and flexible learning. From the survey, it is known that by using the computer as a learning

tool to save time up to 30% of the time needed to learn without the tools.

4. Audio Visual. Uses of audio visual media are usually more attractive and also give a longer memory than just by listening to it because it is considered to be static and boring. By using audio visual material learned cause becomes more attractive.

SYSTEM DESIGN AND IMPLEMENTATION

This CAL applications are made consists of 7 modules such as Stack, Queue, Linear Single Link List, Circular, Single Link List, Tree, and Hashing.

Figure 4 below is a learning module for the stack. In this section, students will see the visualization began when the stack is created, then when entering a data (push), issued a data (pop), and check whether the stack is empty/full. For each process will be demonstrated visually, the program tracing and its watch and also the description of the process.

The first function to be performed is the Create, just as the use of the data structure in which the stack must be initialized before they can be used. Users are asked to enter the size of the tables in this stack, which is the maximum amount of data that can be inserted into the Stack.

After Stack initialization is complete, then the other functions can be used, for example, the Push function. This function aims to enter data into the stack. Users can enter the desired data through a TextBox. If the Stack is full, it will display an error message.

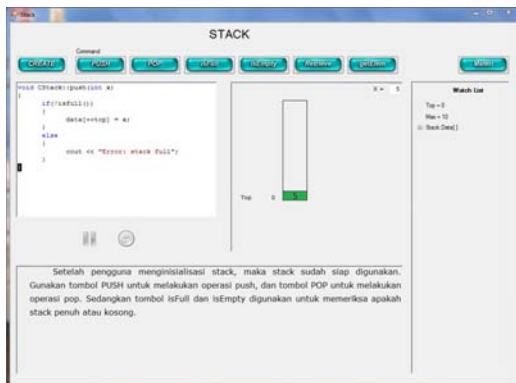


Figure 4. Stack

Figure 5 below is a learning module for the queue. In this section, students will see the visualization began when the queue is created, then when entering a data, issued a data and

check whether the queue is empty/full. For each process will be demonstrated visually, the program tracing and its watch and also the description of the process.

The first function that must be executed is the Create function, the queue must be initialized before they can be used. Users will be asked to enter the size of the tables in this queue. After the initialization queue is finished, the other functions can be used, for example Enqueue function. This function aims to enter data into the queue.

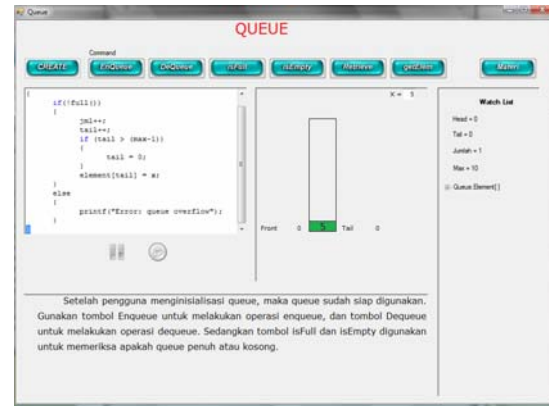


Figure 5. Queue

Figure 6 below is a learning module for the single linked list. This section will explain how to enter data in front/behind/middle, delete data, and searching for a data on a Single Linked List.

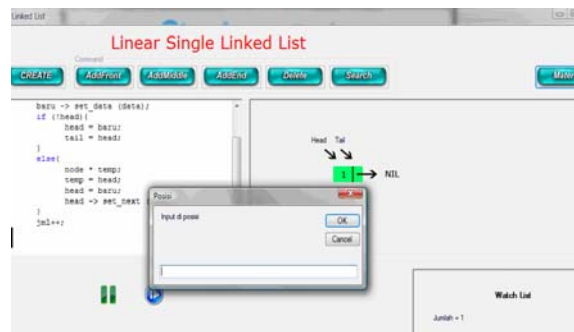


Figure 6. Single Linked List

Figure 7 below is a learning module for the Circular Single Linked List. This section will explain how to enter data in front/behind/middle, delete data, and searching for a data on a Circular Single Linked List.

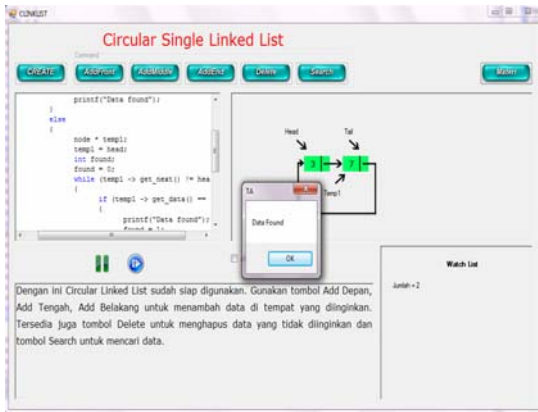


Figure 7. Circular Single Linked List

Figure 8 below is a learning module for the Binary Tree. This section will explain the initialization of the tree, inserting data, deleting data, searching for a data in the binary tree and displaying the content tree in preorder, inorder and postorder.

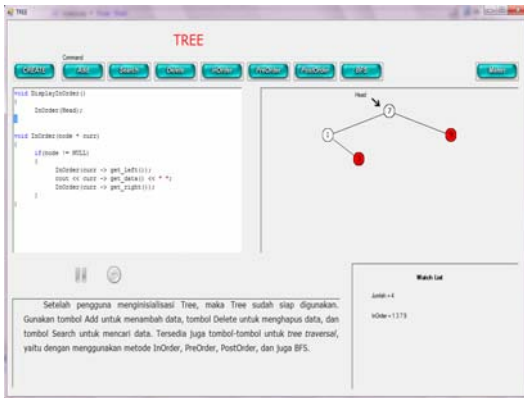


Figure 8. Binary Tree

Figure 9 below is a learning module for Hashing. This application uses Hash Mod TableSize for Hashing Function and LISCH method, Progressive Overflow, or Linear Quotient when collision happened.

To Create feature, users must select the method of collision that want to use it first. After that, the user enters the desired table size.

After Hashing initialization is done, and then the user can run the Add function to add data via InputBox. If data has already been input, it will display the error message.

Search function is used to find data in the Hash Table and used to calculate how much probe is needed to find the data. If the data is not found, it will display an error message.



Figure 9. Hashing

Figure 10 is Recursion module, which in this module there are several examples of recursion cases include Hanoi Tower. The program will show how to solve the case of Hanoi Tower where the number plate based on input from the user.

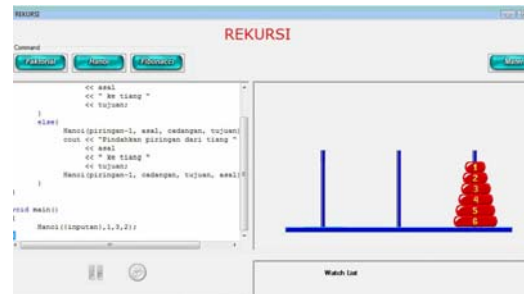


Figure 10. Recursive

CONCLUSION

This application has been tested by 20 students. Based on test results, 75% said that the animation can make them a better understanding of the data structure material, 50% said user interface is very good, and 52.5% said that the program is easy to use.

REFERENCES

Adipranata, R., Liliana and Dewi Indah Sari (2006). Pembuatan Computer Aided Learning Mengenai Hewan Vertebrata bagi Sis wa SMP. Seminar Nasional Teknologi III - Transformasi Teknologi Untuk Peningkatan Kualitas Hidup Manusia, Universitas Teknologi Yogyakarta.

Budhi, G.S. (2008). Penggunaan Strategi Pembelajaran Contextual Teaching and Learning (CTL) Untuk Mencapai Tujuan dari Mata Kuliah Desain dan Analisa Algoritma,

Lehman, T.W. (2005) Data Structure *Visualization as a Learning Tools*.

Weiss, M.A. (1996) Algorithms, Data Structures, and Problem Solving With C++ (2nd ed.). Addison-Wesley

Weiss, M.A. (2003) Data Structure and Problem Solving Using C++ (2nd ed.). Addison-Wesley

