The application of algebraic methods to simplify the more complex type of static data structure through visualization approach

M Hasbi

Department of Mathematics, Hasanuddin University, Jl. Perintis Kemerdekaan, KM. 10, Makassar, Sulawesi Selatan, Indonesia

hasbi.unhas@gmail.com

Abstract. This paper had the result of action research (AR) of Data Structure course. This paper discussed the elaboration of high level complexity of composite type of data structure applying algebraic method through visualization approach (VA). The purpose of this research was to simplify the students understanding at any levels of complexity of static type of data structures through the application of algebraic methods with visualization approach. The roles of the applications were the elaboration of any level of type complexity of data structures logically and systematically at to identify each of their individual access. The method applied to this AR was an algebraic method through visualization approach of composite type of data structures in two cycles. The first cycle applied manually visualization approach, and the second applied simple MS-Powerpoint animation of visualization approach. Some examples of third complexity level of composite type of data structures were presented to demonstrate the methods. The results showed that, before the method is applied, none of the 43 students participating in the course of data structures that can be logically and systematically identifies each individual access from any type of data structure composites tested. However, after the second cycle levels of AR applying VM shown that the average score of learning outcome was 78.00 and classical mastery learning achievement was 76.74%. It meant that the students achievement were very high. The conclusion was the application of VM significantly improved data structure programmer competency.

1. Background

Being a programmer was a profession envisioned by computer science students [1]. A good computer programmer can translate a functionality, that was categorical and hierarchical, in the language of a computer code. “A good programmer understands the deep structure of the program. When he/she speaks it was on behalf of the program. The tests and proofs of truth were Quality Assurance tests and other instruments that measure the speed, efficiency, security, and reliability of the site” [2].

A preliminary survey through competency test of a depth of three levels of complexity of data structure which conducted to the 43 students of 5th semester of mathematics department who were taken data structure course shown that only six (14%) students who could elaborate data structure with 1st-level
complexity, only three students (7%) who could elaborate data structure with 2nd-level complexity, and no students who could elaborate data structure with 3rd-level complexity.

2. Literature review

Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage [3].

A data structure is a specialized format for organizing and storing data. General data structure types include the array, the file, the record, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways. In computer programming, a data structure may be selected or designed to store data for the purpose of working on it with various algorithms [4].

Computer programming was the integration of data structures and algorithms. In other words, the programming language consists of two main parts, namely the data structures and algorithms part. The data structure stated as a variable in the declaration part of a program, while the algorithms stated in statement part of a program.

The data structure has its own data type. The data type can be a standard type (primitive-type) or a composite data type that had to be specified by user (user-defined-type). The Level of complexity of a Data structure determined by the level of complexity of its data type. The higher the complexity level of its data type the more complex the data structure was. Therefore, the data types of such data structures must be user-defined on the basis of primitive data types [5]. A composite data type was one in which each value was a collection of component items. The entire collection was given a single name that has its primitive type, yet each component can still be accessed individually [6]. Each of the component item of the collection that already had single name had its own primitive type. For example:

```
Type X = Record
  a: Integer;
  b: String[10];
end;

Var Y, Z: X;
```

The X-type was an user-defined-type in PASCAL. The X-type was a composite type that consist of collection of two component items, i.e. a-component of integer type and b-component of String[10] type. The single names of the collection of componen items were X.a and X.b. The name of the X-type called formal-name, and the names X.a and X.b were called actual-names. Each of the actual-names, such as X.a has a simple type Integer, and X.b has a simple type String[10].

2.1. Formal VS Actual Data Structure

A data structure was declared as a variable (Var <identifier-name> in PASCAL language). Both Y and Z above were data structure identifiers with an X-type. Because The X-type was still not primitive-type level, so the data structure identifier Y or Z of X-type were still not recognized by an algorithm directly. An algorithm only recognized a data structure identifier that had a primitive type (Jaini & Rama, 1st Ed. 2015). In order the data structure identifier Y or Z could be recognized by an algorithm, then the data structure Y or Z should be elaborated systematically until they had primitive-types. For these cases, the primitive-types of Y were Y.a of type integer and Y.b of type string[10], and the primitive-types of Z were Z.a of types integers and Z.b of type string[10].
the method to elaborate systematically to find out the primitive-type of any complexity-level of data structure identifier were applying an algebraic methods to simplify the data structure through visualization approach in stages, as discussed in this paper.

2.2. Complexity Level of Data Structure

A data structure constructed based on the requirement of an algorithm. Some of algorithms required simple level of data structure and some others required more complex data structure level. The level of data structure complexity indicated by the level of it's type complexity. For example:

\[
\begin{align*}
\text{Type } X &= \text{Record} \\
& \quad \text{a: Integer;} \\
& \quad \text{b: String[10];} \\
& \quad \text{end;} \\
\text{Type } H &= \text{Record} \\
& \quad \text{c: array[1..3] of Real;} \\
& \quad \text{d: Boolean;} \\
& \quad \text{e: String[40];} \\
& \quad \text{end;} \\
\text{Type } K &= \text{Record} \\
& \quad \text{f: X;} \\
& \quad \text{g: H;} \\
& \quad \text{j: Char;} \\
& \quad \text{end;}
\end{align*}
\]

\[\text{Var } Y : K;\]

The \(X\)-type had 1\(^{st}\) level complexity because the primitive type of \(X\) could be identified through one elaboration level. The \(H\)-type had 2\(^{nd}\) level complexity because it had \(c\)-type that had Array-type which had 1\(^{st}\) level complexity. So the whole primitive type of \(H\) could be identified through 2\(^{nd}\) elaboration levels. The \(K\)-type had 3\(^{rd}\) level complexity because it had \(g\)-type that had \(H\)-type which had 2\(^{nd}\) level complexity. So the whole primitive type of \(K\) could be identified through 3\(^{rd}\) elaboration levels. So, the data structure \(Y\) with \(K\)-type had 3\(^{rd}\) complexity level.

2.3. The Algebraic Methods to Elaborate Data Structure In Stages

Elaboration of data structure with algebraic method was a method applied to identify the whole data structure identifiers which has primitive types. As an example:

\[
\begin{align*}
\text{Type } K &= \text{Record} \\
& \quad \text{f: X;} \\
& \quad \text{g: H;} \\
& \quad \text{j: Char;} \\
& \quad \text{end;}
\end{align*}
\]

The \(K\)-type was still not a primitive data type. Therefore, the type of data structure \(Y\) of \(K\)-type was also still not primitive. The \(K\)-type could be found its primitive type by elaborating it algebraically in stages untill the whole data structure identifiers have primitive type. The \(K\)-type was still not primitive type because there were type \(K.f\) was of type \(X\) and \(K.g\) was of type \(H\). By elaborating algebraically, the composit type \(K\) was equivalent to the type below respectively:
Type $K = \text{Record}$

\[
\begin{align*}
\text{f.a: Integer; } & \\
\text{f.b: String[10]; } & \\
\text{g.c: array[1..3] of Real; } & \\
\text{g.d: Boolean; } & \\
\text{g.e: String[40]; } & \\
\text{j: Char; } & \\
\end{align*}
\]

end;

So, the data structure identifier $Y$ ($\text{Var } Y: K$) was equivalent to the data structure identifiers with primitive type below:

\[
\begin{align*}
\text{Var } Y.\text{f.a: Integer; } & \\
\text{Var } Y.\text{f.b: String[10]; } & \\
\text{Var } Y.\text{g.c[1]: Real; } & \\
\text{Var } Y.\text{g.c[2]: Real; } & \\
\text{Var } Y.\text{g.c[3]: Real; } & \\
\text{Var } Y.\text{g.d: Boolean; } & \\
\text{Var } Y.\text{g.e: String[40]; } & \\
\text{Var } Y.\text{j: Char; } & \\
\end{align*}
\]

2.4. Visualization Approach to Elaborate Data Structure

“These are methods to elaborate (mostly) qualitative concepts, ideas, plans, and analyses through the help of rule-guided mapping procedures. In Concept Visualization knowledge was usually presented in a 2-D graphical display where concepts (usually represented within boxes or circles), connected by directed arcs encoding brief relationships (linking phrases) between pairs of concepts. These relationships usually consist of verbs, forming propositions or phrases for each pair of concepts” [7]. “Visualization has proven to be an effective strategy for supporting user in coping with complexity in knowledge- and information-rich scenarios” [7].

This visualization method was applied to visualize simple or composite type of data structure concepts of a computer program. The following were some examples of simple or composite type of data structure:

| Declaration | Visualisation |
|-------------|---------------|
| \text{Type } X: \text{Integer;} | ![Diagram of X] |
| \text{Type } Y: \text{Char;} | ![Diagram of Y] |
| \text{Var A: } X | ![Diagram of A] → ![Diagram of Integer] |
Type $P = \text{Record}$

\begin{align*}
 & \text{a: Integer;} \\
 & b: \text{String[10];} \\
\end{align*}

\text{end;}

\text{Var Y, Z: P;}

For example. Create a data structure to store fifty data of Hasanuddin University’s students, with the data attributes were: \text{Student Identification Number, Name, Date of Birth, Address, and Gender.}

To construct easily the type of data structure of the problem were: The first, visualize the data structure of this problem, for example: \text{student-i}, where \(i = 1, 2, 3, \ldots, 50\): Nim-i for \text{Student Identification Number, nama-i for Name, Tgl_lhr-i for Date of Birth, Alamat-i for Address, and Jk_i for Gender.}. The second, identify the type of data structure that corresponded to each attribute. From the example above, it shown that each student stored 5 groups of different data types. \text{NIM-i requires String[9]; Name-i requires String[50]; Date-i require Date; Address-i requires String[100]; Jk-i requires Char.} Because of each five-data-attributes of student can not be separated from one another, so the type of data structure that can be applied was a record type (PASCAL Declaration type) or Struct (C type declaration). Here the PASCAL type declaration.

3. Research Methodology

This research applied an action research (AR), which consists of two 2 cycles based on [8] concepts. First cycle was manually visualization method (MVM), and the second was simple MS-Power Point animation of visualization method (SPAOVM). Those two cycles consisted of four phases: planning, implementation, observation and reflection.

The research subjects were students of data structure course, Mathematics study program, department of Mathematics, Hasanuddin University, Makassar. This AR involved 43 students who taken data structures course at the fifth Semester, and taken place at the beginning of 2015/2016 semester. The object of this research was visualization methods of composite type of data structure concept of data structure programming languages, such as Pascal, C, or Java.
The goal was to improve students understanding of basic concepts of data structure programming languages through a visualization method. Therefore, there were some stages in this AR method. The stages were:

3.1. Action Plans

3.1.1. Planning
At this stage, researchers explain some steps to be followed as follows.
   a. Designing for 2 meetings in two weeks (about 4 hours face to face lectures, 4 hours of structured task, and 4 hours of independent tasks).
   b. Preparing learning plans of some algebraic and visualize data structures.
   c. Setting up some examples of algebraic and visualize data structures, ranging from level-1 sd. Level-3.
   d. Setting up some exercises and competency tests of algebraic and visualize data structures, ranging from level-1 until level-3.

3.1.2. Actions
At this stage, researchers explain the application of manually visualization method. Subsequently, researcher demonstrated, in stages, some examples of applying manually visualization method. The students could ask questions during the visualization process in order to ensure that all students understand the method correctly.

After that, researchers gave the students exercises. Researchers accompanied with the students to ensure that all students were able to complete their own exercises. Finally, the researchers gave the competency test in the form of assay.

3.1.3. Observation
Aspects observed in this AR were:
   a. The truth of data structure in accordance with the planned problems defined by the students.
   b. The truth algebraic application with visualization method of the planned exercising data structure
   c. The level of active participation of the student in the learning process.
   d. The level of motivation of the student in the learning process.

3.1.4. Reflection
Reflection was conducted based on the problems faced by the student during the observation, includes:
   a. The student problems during exercising the planned problems.
   b. The student problems of activeness in the learning process.
   c. The student problems to completed their task.

From these data, researchers examined the successes and failures based on the purpose of this AR. The reflection results become the basis for determining the next action in the second cycle in order to achieve the classical mastery learning of the AR.

3.2. Research Data
Data to be collected:
   a. The student competency to identify the primitive type of the third complexity level.
   b. The student competency to apply VM.
   c. The students participation in learning process.
   d. The students motivation in learning process.
3.3. **Data collection Instrumenten**

Instrumenten to be implemented:

- **a.** Test methods. This method was performed to determine learning outcomes of students after completing the learning by applying this concept visualization methods.
- **b.** Observation method. This method is applied to measure success indicators of the learning process. The detailed observation sheet form displayed some aspects of the process that must be observed. Researchers acted as an observer. Researchers observed the learning process from start to finish.
- **c.** Questionnaire Method. The questionnaire method was provided for students to know the motivation of the students with the application of the visualization methods of composite type of data structure concept.
- **d.** Document methods. This method was applied to document all problem solving results in the form of homework.

4. **Data Analysis Metode**

4.1. **Learning outcomes**

Learning outcomes data obtained through competency test in the form of essay questions applying a scale of 0 s / d 100 and analyzed in two levels. The first level was to calculate the average score of student’s competency test scores and classical mastery learning achievement of all students. The second level was to analyze the improvement of students competency and classical mastery learning achievements from the first cycle to the second cycle. Therefore, the data analysis was done both quantitatively and qualitatively classical mastery learning achievements were measured by the following formula:

\[
Clc = \frac{NS}{N} \times 100\%
\]

Where:
- \(Clc\): classical learning completeness
- \(NS\): Number of student who had competency test scores \(\geq 70\)
- \(N\): Number of students

4.2. **Students Response**

The questionnaire prepared using five alternative answers applying a Likert scale models: Strongly Agree (SA), Agree (A), Hesitate (H), Disagree (DA) and Strongly Disagree (SDA).

On the positive statements, respondents answered: strongly agree scored 5, agreed scored 4, hesitant scored 3, disagree scored 2 and strongly disagree scored 1. On the negative statement, respondents answered: strongly agree scored 1, agree scored 2, hesitated scored 3, disagree scored 4 and strongly disagree scored 5.

4.3. **Learning outcome Indikators**

Learning outcome indicators of this AR were measured immediately after the second week of the learning process were performed:

- **a.** Learning outcome were considered successful if the students achieved an average grade of at least 75 and classical mastery learning of at least 75% or high category [9].
- **b.** The response of students to the application of visualization methods were at least good.
5. Result and Discussion
The data of AR results were obtained from the competency test results and the completeness of learning achievements result. Data from this AR were grouped into two parts. The first data part was the result from the first cycle, and the second data part was the result from the second cycle.

5.1. Result of Action Research of The First Cycle
5.1.1. Learning outcome
The results of data analysis showed that the average learning outcome of the competency test was 49.63 of 100 and the completeness of learning achievement was 51.16%. This meant that the learning outcomes had not reached the minimum students competency and classical mastery learning criteria. It also meant that most students scored below the minimum criteria.

5.1.2. Student Responses
From the results of the questionnaires distributed to the students, to determine their response to the application of visualization methods manually, indicated that the achievement was good. This was shown by the average score of the questionnaire 4.01 of 5.00.

5.1.3. Reflection
The reflection result on the first cycle was found that some students did not understand in detail the application of manually visualization method. They needed more repetitions of the given examples. In addition, they require repetition at any times of these methods if they have learning difficulties. Therefore, manually visualization methods in the first cycle need to be improved in the second cycle. For that reason, the visualization method in this second cycle applying simple MS-Powerpoint animation of visualization method of composite type of data structure concept. With the implementation of simple animation method, then the student can repeat any example visualization according to their needs.

5.2. Result of Action Research Result of Second Cycle
5.2.1. Learning outcome
From the competency test results, students obtained an average score in the second cycle was about 7.60 of 100. There were about 10 of 43 or 23.26% of students had not yet achieved their classical learning completeness, and about 33 of 43 students or 76.74% of students had achieved their classical learning completeness. It meant that the students had reached their competency and classical mastery learning criteria. It was convinced that with the help of simple computer animation improved learning outcome of the data structure programmer. It was because of the simple MS-Powerpoint animation will facilitate their own learning process any time, any where, and any how they needed.

5.2.2. Student Responses
From the results of the questionnaires distributed to the students, to determine their response to the application of visualization methods indicated that the achievement was good. This was shown by the average score of the questionnaire 4.23 of 5.00.

5.2.3. Reflection
The result of the students achievement in the second cycle has reached an average of 7.20 and classical mastery learning had reached an average of 76.74%. This meant that a student has reached a predetermined success criteria at the beginning of this AR of 6.80 and classical mastery learning 75%. So no need to proceed further on the next action cycle.
5.3. Discussion

5.3.1. Learning Process Activity

The learning process begins with an explanation of the application of visualization methods of the type of data structures concept to the students. Learning atmosphere with applying visualization method of composite types and data structure concept seemed very serious. All students follow strictly all stages of visualization methods were discussed by the lecturer. Students immediately asked what they did not understand at the time of the visualization method in progress. Beginning with the explanation of the application of visualization methods and data structures of type concepts to students.

Afterward, all the students did practice questions given to him without any fear and distress. Students were able to answer the exercises correctly look happy at all. While the students were not able to answer the exercise correctly was very curious. Some of them asked his close friend, the others asked her professor about how.

Learning atmosphere was a bit crowded, but it did not because they were joking or playful each other, but even more because of their curiosity to solve the exercises. There were some difficulties experienced by the students in solving exercises, it was because they were less familiar in elaborating algebraically with the type an data structure identifiers to achieve the simple type of data structure identifier.

Students perform with enthusiasm and active learning according to the learning objectives. The success rate of the learning process was not only measured by the extent to which students have mastered the subject matter, but also measured by the extent to which students attend a learning process. Lecturers did not only serve as a source of learning, but also serves as a mentor and facilitator so that students were willing and able to learn on their own. Students were not considered as learning objects that can be regulated and limited by the willingness of faculty, but students were placed as a subject of study according to their talents, interests and ability.

5.4. Learning outcome

After the learning process was complete, the lecturer did achievement test to measure the ability of students' understanding of the concepts they have acquired in the form of assay testing. In the first cycle, students achievement an average score was equal to 49.63 of 100 and completeness of learning achievement was equal to 51.16%. Meanwhile, in the second cycle, students gained an average score equal to 7.60 and completeness of learning achievement was equal to 76.74%. Thus, student learning outcomes have increased an average of 2,87 and completeness of learning outcomes increase 25.58%.

6. Conclusion

Application of visualization method of composite type of data structure concept, applying simple animation of MS-Powerpoint, improved student learning outcome in identifying primitive type of composite type with complexity level-n, systematically, logically, and easily. It is shown that at the end of the first cycle, the average value of student results was 49.63 of 100, and the learning classical mastery learning of the classroom was 51.16%. At the end of the second cycle average score of learning outcomes was significantly increased to 7.80 (or an increase of), and classical mastery learning class was 76.74% (or an increase of 25.58%).
References

[1] Oktavian, D. P. (2010). *Menjadi Programmer Jempolan Menggunakan PHP*. Jakarta: PT Suka Buku.

[2] Girard, M., & Stark, D. (2002, Juni -). *Distributing intelligence and organizing diversity in new media projects*. Retrieved Oktober 28, 2016, from Sociedade e Estado: http://www.scielo.br/scielo.php?pid=S0102-69922002000100009&script=sci_arttext&tlng=pt

[3] Studytunight.com. (2017, - 18). *Introduction to Data Structure*. Retrieved from Data Structure: http://www.studytunight.com/data-structures/introduction-to-data-structures

[4] TechTarget. (2017, - -). *Definition of Data Structure*. Retrieved from SearchSQLServer: http://searchsqlserver.techtarget.com/definition/data-structure

[5] Jaini, C. V., & Rama, S. S. (1st Ed. 2015). *Computer Applications A Textbook for Class X*. New Delhi 110-014: Madhubun.

[6] Dale, N., Weems, C., & Headington, M. (2003). *Introduction to Java and Software Design*. United States Of America: Jones and Bartlett Publisher. Inc.

[7] Keller, T., & Tergan, S.-O. (2005). *Knowledge and Information Visualization*. Berlin: Springer.

[8] Kemmis, S., & McTaggart, R. (1988). *The Action Research Planner (3rd ed).*. Geelong, Australia: Deakin University Press.

[9] Mukminan. (2003). *Pembelajaran Tuntas (Mastery Learning)*. Jakarta: Departemen Pendidikan Nasional, Ditjen Dikdasmen, Direktorat PLP.