ATLANTIS PRESS

Advances in Social Science, Education and Humanities Research, volume 242

2nd International Conference on Vocational Education and Training (ICOVET 2018)

Implementation of *Breadth First Search* Algorithm for Verification in the Informatics Engineering Education (PTI) UM

Syaichul Fitrian Akbar[1]
Students of Vocational Education, Graduate School, Malang State University
Malang – Indonesia
sfaregs@gmail.com

Tito Tri Prabowo[2]
Students of Vocational Education, Graduate School, Malang State University
Malang – Indonesia
titoprabowo@hotmail.com

Rita Purniawati[3]
Students of Vocational Education, Graduate School, Malang State University
Malang – Indonesia
rita.cantik@yahoo.co.id

Anik Nur Handayani[4]
) *Lecturer of Vocational Education, Graduate School, Malang State University,*
Malang – Indonesia.
aniknur.ft@um.ac.id

***Abstrak*—Courses are the requirement in completing a study; students are considered to have completed their learning process if they have passed all courses determined by the competence of the study program. Students who are yet to pass their credit (SKS) mean that they have not completed their study. Thus, a method which can be used to verify whether students have taken or passed the predetermined courses, one of which, is Breadth First Search (BFS). Such a method conducts a left-to-right screening process to search and verify the courses with algorithm, pseudocode, and graph being the implementation result of the BFS method.**

In terms of searching, there are four parameters involved: time, memory, complexity, and optimality. Based on the BFS method implementation, it is known that the time spent for the implementation is relatively short, since the BFS generated every possibility collectively, thus calculating the distance of each node is not necessary. The memory required is relatively small, since the BFS method is a blind search. The complexity is in the complete category, since all of the existing possibilities were generated followed by the left-to-right searching process. In terms of optimality, further research using different method was required for comparison.

Keywords—Breadth First Search; Course Verification; PTI UM;

I. INTRODUCTION

In the scope of education, especially higher education, the main requirement related to the completion of an educational program is to pass the likes of predetermined courses, industrial practices, school introduction, and thesis. A number of requirements possibly affect the result and determination of graduation or completion of the study period with courses used as the requirement for study completion being the example, since students are required to pass the courses as a form of completion based on their study program. If the students fail, the credit (SKS) is yet to be met, thus they cannot graduate.

For example, a student, who has passed the courses, fieldwork practice, and thesis, proceeds to graduation, but later finds out that the student is yet to take one remaining course. Such a case is most likely to occur that an information verification process related to whether or not students have passed all of the required courses is required. The Breadth First Search algorithm can be used as the method to verify the courses. Breadth First Search is an algorithm to conduct expansive data searching process from left to right on each node in pre-order manner.

This research aims to determine the possibility of the BFS algorithm to be applied in the process of course verification, which in turn, will facilitate the verification process related to the courses taken by the students.

II. THEORETICAL FRAMEWORK

According to Suyanto (2014), BFS is a searching method carried out sequentially on all nodes at each level from left to right. If a solution has not been identified at one level, the search continues to the next level, and so on until the solution is identified. The BFS ensures the capability to identify the best solution possible (if any). In other words, the BFS is complete and optimal (Suyanto, 2014). Prasetiyo and Hidayah (2014) state that Breadth First Search is an algorithm conducting expansive search, investigating node in pre-order manner by visiting one node then visiting all nodes adjacent to the node; followed by, visiting the unvisited nodes and adjacent to the previously visited nodes, and so on. According to Budianto, Lasguido, Fathurrahman, and

Wibowo (2011), the BFS is a simple algorithm used on graphs.

The BFS method will search the threshold between visited nodes and unvisited nodes, so that each node being at s distance from the initial node will be visited before visiting a node that is at s + 1 distance from the initial node. Graph search on BFS is conducted by searching each node starting from the initial node. The queue data structure is used in the BFS expansion process. Queue is used to hold each node or a possible valid node to pass. The nature of queue is *First In First Out* (FIFO), so that each node firstly selected and expanded by BFS will be first processed. Such a nature of queue maintains the nature of the BFS algorithm that is to search the order layer. This algorithm also requires a boolean table to store the visited nodes, so that no node is visited more than once.

The BFS method workflow is to store the visited node in a queue, which refers to the adjacent nodes, and will be visited according to the queue order. To clarify the workflow of the BFS algorithm, the following is the elaborative procedure of the BFS algorithm: (1) Insert the end node (root) into the queue; (2) Take the node in the beginning of the queue and check whether or not the node is a solution; (3) If the node is a solution, the searching process is deemed complete and the results are returned; (4) If the node is not a solution, insert all nodes adjacent to the latter node into the queue; (5) If the queue is empty and each node has been checked, the searching process is deemed complete and the result are returned without any solution identified; and (6) Repeat step (2) until the solution is identified.

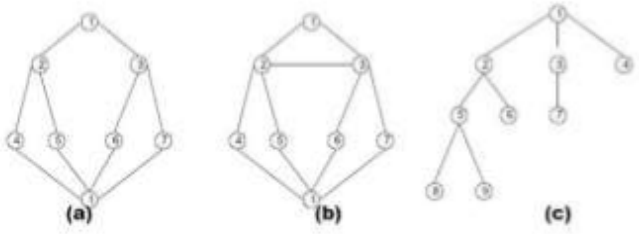

Figure 1. Illustration of BFS Algorithm

The description of each figure is as follows; (1) Figure 1.a is in the order of: 1, 2, 3, 4, 5, 6, 7, 1; (2) Figure 1.b is in the order of: 1, 2, 3, 4, 5, 6, 7, 1; (3) Figure 1.c is in the order of: 1, 2, 3, 4, 5, 6, 7, 8, 9. There are 4 searching parameters: time, memory, complexity, and optimality.

III. RESEARCH METHOD

Figure 2 is divided into 8 columns comprising 1st to 8th semester, also indicates the relationship between Scientific Compulsory Course (MWK) and prerequisite courses marked by a connecting line. In semester 1, all courses are MWK. Subsequently, the courses in the first semester are the prerequisite course in the following semester, and so on.

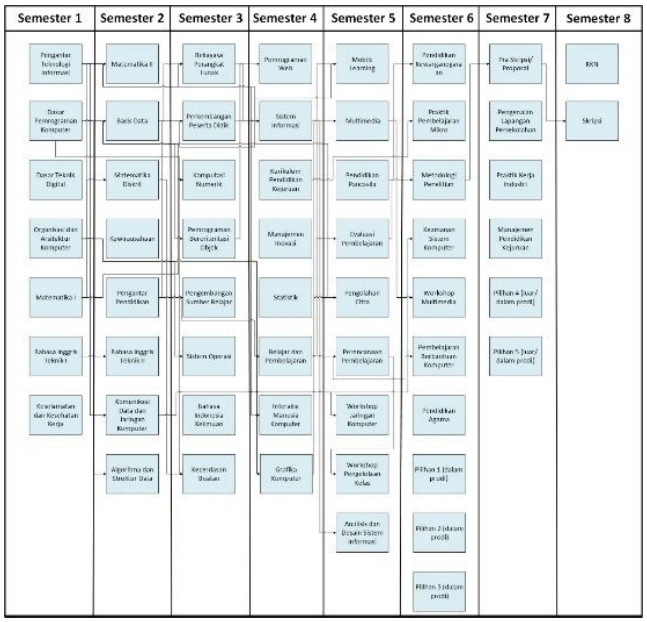

Figure 2. Block Diagram of Courses in Informatics Engineering Education (PTI) UM 2018

For instance, the Introduction to Information Technology course is the prerequisite of the Data Communication and Computer Network in semester 2; the Software Engineering in semester 3; and (1) the Information System and (2) the Human and Computer Interaction in semester 4.

The Basic Computer Programming course if the prerequisite of (1) the Data Basis and (2) the Algorithm and Data Structure in semester 2; (1) the Numeric Computation, (2) the Object-Oriented Programming, and (3) the Artificial Intelligence in semester 3; and the Imagery Management in semester 5.

The Computer Organization and Architecture course is the prerequisite of the Data Communication and Computer Network in semester 2; the Operating System in semester 3; and the Computer Graphic in semester 4.

The Mathematics I course is the prerequisite of (1) the Mathematics II and (2) Discrete Mathematics in semester 2; the Numeric Computation in semester 3. The English for Engineering I course is the prerequisite of the English for Engineering II in semester 2.

The remaining two courses in the first semester; the Basic Digital Engineering and the Occupational Health and Safety, are not the prerequisites of any courses meaning that those course cannot be broken down any further.

Upon the analysis of courses in the first semester (column 1), courses in the second semester (column 2) were analyzed. The Data Basis course is the prerequisite of the Software Engineering in semester 3; the Information System in semester 4.

The Discrete Mathematics course is the prerequisite of the Artificial Intelligence in semester 3. The Introduction to Education course is the prerequisite of (1) the Student Development and (2) the Teaching Material Development in semester 3; the Learning and Teaching in semester 4. The Data Communication and Computer Network course is the prerequisite of the Computer Network Workshop in semester 5; and the System Security in semester 6.

The Entrepreneurship course is not the prerequisite of any courses meaning that it cannot be broken down any further. The Mathematics II, English for Engineering II, and Algorithm and Data Structure courses have their prerequisite in the previous semester, but are not the prerequisite of any courses in the following semester meaning that the courses cannot be broken down any further.

Upon the analysis of courses in the second semester (column 2), courses in the third semester (column 3) were analyzed. The Software Engineering course is the prerequisite of the Information System in semester 4.

The Object-Oriented Programming course is the prerequisite of the Web Programming in semester 4; the Mobile Learning in semester 5.

The Indonesian for Scientific Purposes course is not the prerequisite of any courses meaning that it cannot be broken down any further. The Student Development, Numeric Computation, Teaching Material Development, Operating System, and Artificial Intelligence have their prerequisite in the previous semester, but are not the prerequisite of any courses meaning that those courses cannot be broken down any further.

Upon the analysis of courses in the third semester (column 3), courses in the fourth semester (column 4) were analyzed. The Information System course is the prerequisite of the Information System Analysis and Design in semester 5.

The Vocational Education Curriculum course is the prerequisite of Microteaching Practice in semester 6. The Learning and Teaching course is the prerequisite of (1) the Mobile Learning, (2) Teaching Evaluation, and (3) Teaching Planning in semester 5; and the Computer-Assisted Teaching in semester 6. The Computer Graphic course is the prerequisite of (1) Multimedia and (2) Imagery Management in semester 5.

The Innovation and Statistical Management course is not the prerequisite of any courses meaning that it cannot be broken down any further. The Web Programming and Human and Computer Interaction have their prerequisite in the previous semester, but are not the prerequisite of any courses meaning that they cannot be broken down any further.

Upon the analysis of courses in the fourth semester (column 4), courses in the fifth semester (column 5) and the remaining semester were analyzed.

IV. RESULTS AND DISCUSSION

The following are graphs produced based on the block diagram in figure 2:

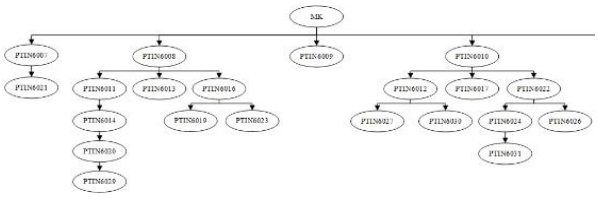

Figure 3 Graph Part 1

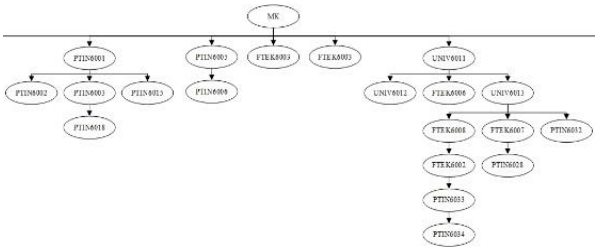

Figure 4 Graph Part 2

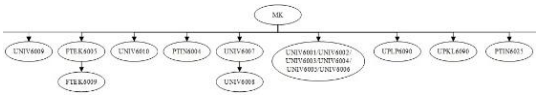

Figure 5 Graph Part 3

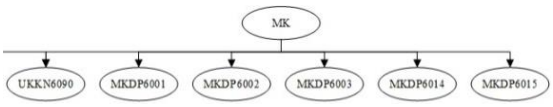

Figure 6 Graph Part 4

Figure 3 to 6 is a series of graphs. Based on figure 3 to 6, then they are known parent node, child node level 1, child node level 2, child node level 3, child node level 4, child node level 5, and child node level 6. The BFS method will search for left nodes to right node until the nodes at that level are totally executed. If there is still no solution found yet, it will be continued to the next level node until the solution is found.

In this case, if there is a course that has not been taken yet, then it is considered as complete or out since there are still courses that have not been taken yet. While the complete requirements are all the courses presented have been taken and have fulfilled the predetermined credits.

The following is a pseudocode of implementing the BFS method:

```
procedure BFS(input v: integer){
    Transversal graf dengan algoritma pencarian
    BFS.
    Input: v adalah simpul awal.
    Output: simpul yang telah dikunjungi
}
Deklarasi
    w : integer
    q : antrian

procedure BuatAntrian(input/output q: antrian){
    Membuat antrian kosong, kepala(q) diisi 0
}
procedure MasukAntrian(input/output q: antrian,
input v: integer){
     Masukkan v ke dalam antrian q pada posisi
     belakang
}
procedure HapusAntrian(input/output q: antrian,
output v: integer){
     Menghapus v dari kepala antrian q
}
function AntrianKosong(input q: antrian) ->
boolean{
    true //jika antrian q kosong
    or
    false //jika antrian q tidak kosong
}
Algoritma
BuatAntrian(q){
     buat antrian kosong
```

```
}
write(v) //cetak simpul awal yang dikunjungi
dikunjungi[v] <- true //simpul v telah
dikunjungi, flag=true

MasukAntrian(q, v){
    Masukkan simpul awal kunjungan ke dalam
    antrian.
    Kunjungi semua simpul graf selama antrian
    belum kosong.
}
while not AntrianKosong(q) do
HapusAntrian(q, v) //simpul v telah dikunjungi,
hapus dari antrian

for (tiap simpul w yang bertetangga dengan
simpul v) do
if not dikunjungi[w] then
    write(w) //cetak simpul yang dikunjungi
MasukAntrian(q, w)
```

The following is the implementation of the graph declaration in the array:

```
$ar[1]['value']      = 1;
$ar[1]['parent']     = 0;
$ar[1]['nama']       = 'Pengantar Teknologi
Informasi';
$ar[1]['prasyarat'] = '-';
```

The graph declaration in the form of array above uses PHP programming language. The first is to initialize the variable ar. The ar variables are identified by several identifiers such as values, parent, name, and prerequisites. In the part of value 1 shows that the ar variable is the first array. The parent section valued 0 is indicating that the array is a parent (not child) or as a basis of the course. The names section is used as the name of the course. The prerequisites section is used for the courses which has prerequisites for the other courses. Likewise, if there is a prerequisite course, then the parent is filled with the value that becomes the prerequisite course as follows:

```
$ar[29]['value']     = 29;
$ar[29]['parent']    = 1;
$ar[29]['nama']      = 'Interaksi Manusia
dan Komputer';
$ar[29]['prasyarat']= $ar[1]['nama'];
```

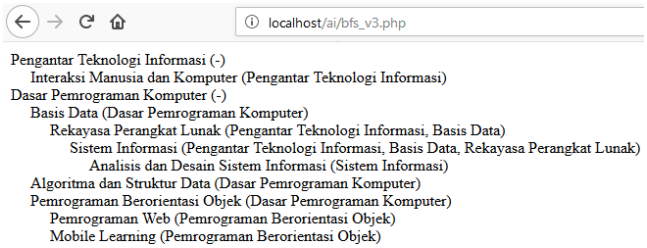

Figure 7 Output Method Implementation

Based on Figure 7, nodes close to the left edge are nodes that have no preconditions. While the sub nodes are child nodes. Therefore, the more it gets into the sub, the higher the level of the child.

V. CONCLUSIONS AND SUGGESTIONS

This research concludes that the Breadth First Search (BFS) method can be implemented to verify passed courses. This BFS method plays a role in analyzing thoroughly or verifying the courses passed by students. With the overall analysis, it is expected that all courses can be detected, so that there is no miss communication when the students processing to course verification.

Since a university certainly has many students, this BFS method becomes a solution that can be implemented in the information system which analyzing courses that have been or have not been passed by students, hence it is more efficient when verifying before students proceed to course verification.

There are four parameters in *searching* namely time, memory, complexity and optimality. Based on the implementation of this BFS method, it can be seen that the time aspect used is relatively fast, because in BFS searches it is generated to all possibilities that no distance calculation is needed. In addition, this method can be combined with the existing system. Accordingly, when it is generated, it can be done based on the course transaction table in each of the courses passed by the student. In the *memory* aspect, the required memory is relatively small because the BFS search method is included in the blind search. In the aspect of *complexity*, it is categorized as complete, because all possibilities are in *generate*, so that all possibilities are searched from left to right. Whereas in the aspect of optimality, further research is needed to be compared with other methods, for example compared to using the DFS method, compared to the combination between the BFS and DFS methods, or compared to other search methods.

Based on the BFS method analysis, further development related to the implementation of this BFS method on the code program (coding) and its validation is still needed. Therefore, this BFS method can be integrated with an information system in the future, for example in the Academic Information System so that it can be used to verify the courses that have been passed or that have not been passed before students proceed courses verification, for example in the form of status or notification that appears in the system.

Suggestions are given to other researchers so that they can further develop integrating the BFS method with the existing information systems. Consequently, the validity test of the implementation of this BFS method on information system such as academic information system can be conducted. In addition, it is suggested to search using other methods, such as DFS or even combine this BFS method with the DFS method, so that the *search* becomes more flexible, not only from left to right, but also it can be done in terms of depth.

REFERENCES

[1] Budianto, Lasguido, Fathurrahman, F., dan Wibowo, A. 2011. *Implementasi Algoritma Breadth First Search dan Obstacle Detection dalam Penelusuran Labirin Dinamis Menggunakan Robot Lego. Jurnal Ilmu Komputer dan Informasi, 4(1),* 15-22. (*online*) (http://jiki.cs.ui.ac.id/index.php/jiki/article/view/153), diakses 20 September 2018.

[2] Haryansyah. 2014. *Terapan Sistem Kecerdasan Buatan Pada Sistem Informasi Akademik berbasis SMS Gateway Menggunkan Metode Breadth First*

Search. *Jurnal STMIK AMIKOM*. Yogyakarta.

[3] Hidayatullah, A. A., Handayani, A. N., dan Fuady, M. J. 2017. *Performance Analysis of Dijkstra and A* Algorithm to Determine Shortest Path of Hexapod Fire Fighting Robot*. *The 7th Annual Basic Science International Conference*. 161-164.

[4] Junaedi. 2010. *Perancangan Penjadwalan Perkuliahan dengan Menggunakan Pendekatan Constraint Satisfaction Problem (CSP) dan Artificial Bee Colony Algorithm*. *Jurnal Universitas Diponegoro*. Semarang.

[5] Adiputra, M. A. 2018. *Penerapan Algoritma BFS dan DFS untuk Penjadwalan Studi*. *Jurnal Intitut Teknologi Bandung*. Bandung.

[6] Prasetiyo, B. dan Hidayah, M. R. 2014. *Penggunaan Metode Depth First Search (DFS) dan Breadth First Search (BFS) pada Strategi Game Kamen Rider Decade Versi 0.3. Scientific Journal of Informatics, 1(2)*, 161-167. (*online*) (https://doaj.org/article/202b39f2ee044a02ad88efbf8e54f44d), diakses 20 September 2018.

[7] Suyanto. 2014. *Artificial Intelligence: Searching, Reasoning, Planning, Learning*. Bandung: Penerbit Informatika.