Journal of Telematics and Informatics (JTI)
Vol.7, No.1, March 2019, pp. 17~27
ISSN:2088-8708

Automatic Lecture Meeting Folder Based on Hierarchical Directories

F.X. Wisnu Yudo Untoro
Informatics Engineering Study Program Faculty of Engineering Wijaya Kusuma University
Jl. Dukuh Kupang XXV/54 Surabaya 60225, tel: +62315677577
e-mail: wisnusakti2410@gmail.com

Abstract

The lecture meeting is a face-to-face meeting between a teacher and the learners in the classroom in the implementation. In the rules that apply that every lecture meeting a teacher must put the signature on the sheet of paper attendance in the signature column. Checking the lecture meeting in this way has its weaknesses, namely easily manipulated. For this reason, there is an idea to design a system that can make automatic college meeting folders every time the teacher runs this system. The experimental results show that the system can automatically create lecture meeting folders for various subject names that are already available.

Keywords*: lecture, meeting, folder, automatic, presence.*

1. Introduction

In teaching and learning activities, there is a term used in the academic environment. Lecture meetings are the terms they mean. Lecture meetings are a form of face-to-face activities between a teacher and students in the classroom in the implementation. At the lecture meeting, a teacher completes the signature column that is available on a piece of paper called the attendance list with the signature printed by the teacher concerned. The printed sign is a proof of the current or already held lecture meeting. An attendance list is an instrument with the benefit of checking the frequency of lecture meetings held by a teacher, but unfortunately, this good instrument often gets less attention.

One of the reasons why attendance lists based on signatures are lacking attention? One of the reasons is that attendance lists are easy to manipulate. For example, when conducting a lecture meeting, a teacher can print his signature in the signature column of the attendance list, more than one column at the same time. The illustration is an overview of events on the attendance list based on signatures that still have weaknesses and have conditions that are easy to manipulate.

As an instrument for checking college meetings, attendance lists play an important role. This attendance list is expected to work well and correctly according to the applicable rules. In simple words, this attendance list requires that teachers be free from all forms of abuse that may occur, such as manipulation or lack of attention. But in reality, the role of attendance at this lecture meeting still cannot be prevented from the form of abuse. From the description, get a conclusion that between the expectations of the attendance list at the lecture meeting with the fact in the field found an apparent gap. For this reason, the idea arises to design a system so that it can automatically make a college meeting folder every time a teacher runs.

To follow-up on this idea, the basic concept of the file system or called file management used in the operating system is then used in this experiment for the framework of thinking. The file system is a method of organizing and retrieving files from storage media, such as hard drives. However, there is another opinion that states that the file system is a storage system that uses a hierarchical directory tree [1]. This opinion is also called the researcher [2], [3], [6]. In the hierarchical directory tree for each directory can still be used as a storage subset directory and directory information tree again [4]. Based on this opinion, decided to design a storage system for this lecture meeting using a file system.

However, the technique used to create a storage directory for lecture meetings still uses the method per lecture meeting made one by one. Whereas in signature-based attendance in its

implementation, it automatically moves to the next signature column. So that this system to work automatically also at the time of making a lecture meeting, the system is given a knowledge [1] [5] [7] [8]. Knowledge given to the system is a tracking knowledge. This tracking knowledge has the task of getting information about the storage of lecture meetings on the drive.

With the design of this lecture meeting folder system, it is expected to be an alternative to replacing attendance based on signatures that are easy to abuse.

2. Research Method

As explained in the previous section, that a lecture meeting is a discussion activity that takes place in the classroom between a teacher and students using face-to-face methods. A teacher who conducts a lecture meeting fills in the signature column on an instrument called attendance based on a signature. There is a weakness based on this signature. The disadvantage lies in the signature column that is easy to manipulate.

The file system used in the operating system is a method used to resolve weaknesses that exist in signature-based attendance by creating a system capable of creating automatic lecture meeting folders based on hierarchical directories (ALMF).

To build an automatic lecture meeting folder system using five stages, namely the system work environment, input data, output, ALMF algorithm, and testing methods. In this section, discuss the five stages one at a time.

2.1 System work environment

This automated lecture meeting folder system is built using the following hardware and software:

2.1.1 Hardware

The hardware specifications used are as follows:

- Processor: Intel (R) Core (TM) i3 3217U CPU @ 1.80GHz 1.80GHz
- Installed Memory (RAM): 2.00 GB (1.89 GB usable)
- System Type: 64-bit Operating System, x64-based processor
- Hard Disk: 500 GB (with four partitions, namely C, D, E, and F)

2.1.2 Software

The software used is as follows:

- Operating system
  The operating system used is Windows 8.1 pro-64-bit. This operating system is a storage place and displays an automatic lecture meeting folder as well as a hierarchical directory tree generated by the system through both file explorer and Command Prompt.
- MATLAB
  MATLAB used is the MATLAB® R2016b version. MATLAB is a programming language used to create programs and test programs from automatic lecture meeting folder algorithms.

2.2 Input data

This automatic lecture meeting folder system is still designed using three types of input data. The three types of input data are as shown in table 1.

Table 1. Input data for the system

Level Directory	Name Level	Selection
1	Drive	C D E F
2	PRESENSI	PRESENSI
3	Courses	Algoritma dan Pemrograman Grafika Komputer Pengolahan Citra Digital

		Kecerdasan Buatan

Table 1 described that all three input data for the system shows level directories. For each level directory, it provides an option as input data. Input data for this system uses the dialog box technique. therefore, the input data for each level directory appears in the dialog box.

2.3 Output

Each system has an output. Even the built automatic lecture meeting folder system also has an output. This system will provide an output in the form of a hierarchical directory tree [1], [4]. In general, an output from this automatic lecture meeting folder system as shown in figure 1.

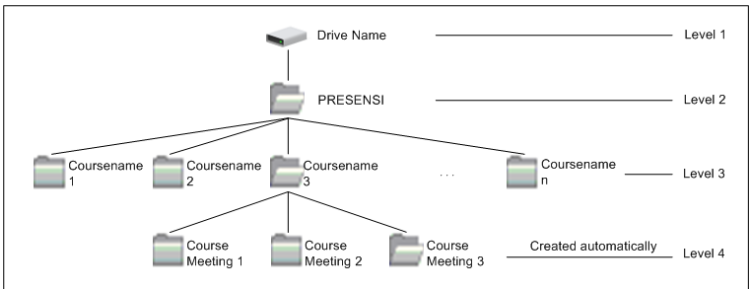

Figure 1. The output from the automatic lecture meeting folder system

Figure 1 describes that the automatic lecture meeting folder system produces an output of a four-level hierarchical directory tree. The first level is the drive, the second level is the PRESENSI folder, the third level is the course folder, and the fourth level is the lecture meeting folder that will be created automatically every time the system runs.

2.4 ALMF algorithm

The algorithm for the automatic lecture meeting folder (ALMF) system is built based on the following opinions: file system [1], [2], [3], [6], directory [4], provide knowledge [5], [7], [8], and algorithm [11]. The results obtained as automatic lecture meeting system algorithms, as shown in Figure 2.

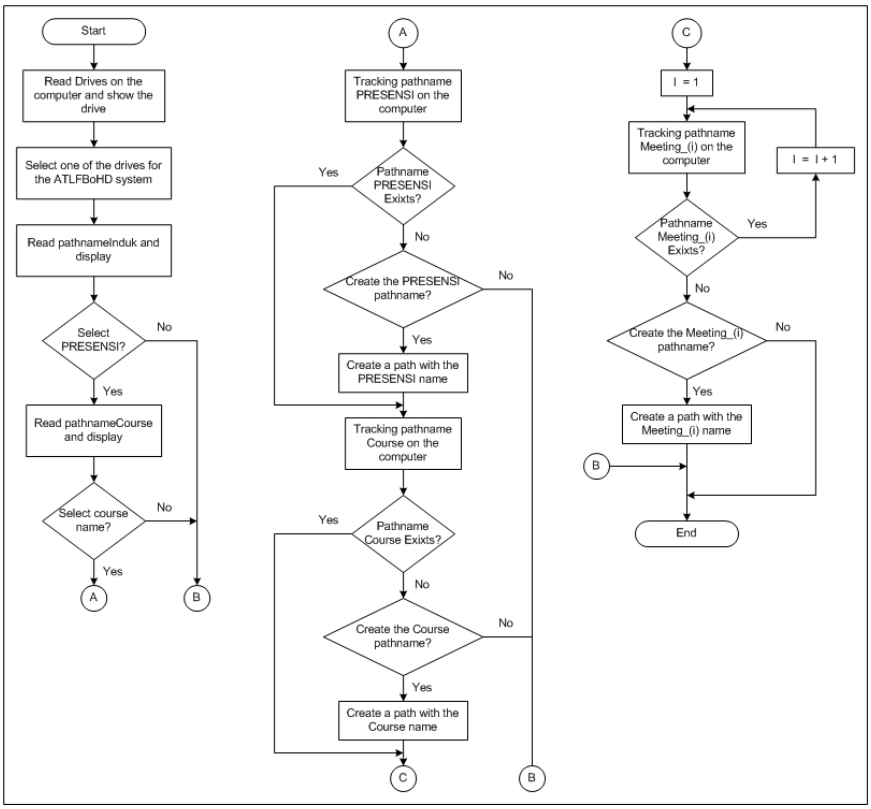

Figure 2. Flowchart system for automatic lecture meeting folder

Figure 2 described that to build an automatic lecture meeting folder system based on this hierarchical directory through step by step as follows:

Step 1: The system reads the drive on the computer and displays all the drives found in the dialog box entry level 1. Here the system gives the user the option to choose which drive to use to build an automatic college meeting folder.
Step 2: The system read and display one PRESENSI name in the dialog box entry level 2.
Step 3: The system read and display the list of courses in the dialog box entry level 3. Here the system gives the user the option to choose one of the course names that appear in the entry dialog box.
Step 4: The system tracks the PRESENSI directory on the computer.
Step 5: The system tracks the course directory in the computer.
Step 6: The system tracks of the automatic lecture meeting folders on the computer

2.5 Testing the ALMF system

In this experiment, testing on an automatic lecture meeting folder (ALMF) system uses two methods, namely white box testing method and black box testing. White box testing method is used to analyze the correct or wrong of a source code program on the system of automatic lecture meeting folders that have been creating. If the module test produces an output that is not as expected, it will return to the source code program to check and repair, which then recompiles to obtain the expected output [9], [10]. While testing black box testing will be used to test the definition of a set of input data conditions and test the functional specifications of the program [10].

3. Results and Analysis

The algorithm of the automatic college meeting folder system has been implemented using MATLAB. The results of the implementation still need to be tested to get information about the output that has been produced as expected or not.

For this purpose, the test will begin by testing the input data for the system [10] and proceed with testing the number of systems run for a course name.

3.1 Testing input data

As explained in section 2.2, the input data for the system uses a dialog box with the name adjusted to the name of the level. There are three types of input data for the system. The first is input data for the drive name. The drive name displayed in the dialog box named entry Level 1 is the names of the drives that are on the computer as output from the process of reading the drive name. In this experiment, the drive name used is drive F:\. The second is the input data for the default name. The default name displayed in the dialog box that is named entry level 2 is the default name as the output of the default name reading process, and the third is the input data for the course name. The subject name displayed in the dialog box that is named entry level 3 is a list of course names as output from the process of reading the course name.

The results of the process of reading the input data by the ALMF system for the drive name, default name, and course name are displayed in each dialog box as shown in figure 1.

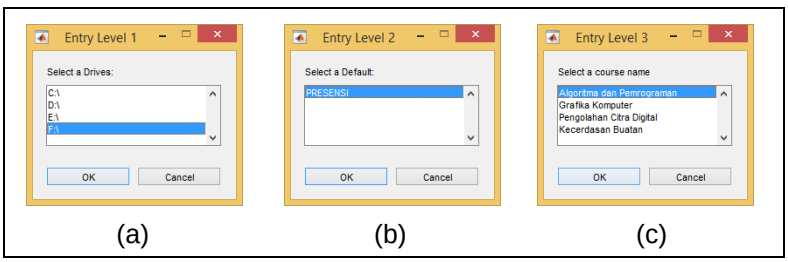

Figure 1. The results of the process read data input (a) drive name, (b) Default name, and (c) course name.

In Figure 1 explains that (a) the results of the system drive read process get four drives in the computer, namely C, D, E, and F. The drive name selected as the input data for the drive name is F. (b) The reading process results the system default name gets one default, namely PRESENSI. So the default name selected as input data is PRESENSI. (c) The results of the process of reading the name of the system course get four-course names. The subject name selected as input data is Algoritma and Pemrograman.

The results of the testing of each input data dialog box are the drive name, default name, and the list of subject names in the ALMF system obtained as shown in table 1.

Table 1. Decision table for input data on the ALMF system

Condition Entries	Drive name	PRESENSI name	Course Name	Expected Result	Result	Action
1	Cancel			False	True	End / Exit
2	Yes			True	True	Proceed to 3
3	Yes	Cancel		False	True	End / Exit
4	Yes	Yes		True	True	Proceed to 5
5	Yes	Yes	Cancel	False	True	End / Exit
6	Yes	Yes	Yes	True	True	Proceed to Info

If the condition entries 6 shown in table 1 are applied using the input data of the course as shown in Figure 1, the system will provide an output in the form of an information dialog box as shown in Figure 2.

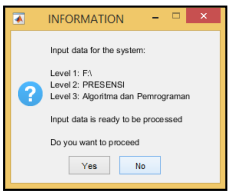

Figure 2. Information on input data that is ready to be processed by the system.

The results of testing the information dialog box in Figure 2, as shown in table 2.

Table 2. Decision table for the information dialog box

Condition Entries	Button		Expected Result	Result	Action
	Yes	No			
1		√	False	True	end / Exit
2	√		True	True	Proceed

Based on the implementation of the tests described above shows that the input data to be processed by the ALMF system has provided the expected output.Where each user (in this case is a teacher) if canceling the input data option, the system will not do any process and exit. However, the opposite will apply if each user agrees to the input data he has chosen. Here the system will use input data as shown in Figure 2 for further processing.

3.2 Testing the ALMF system is run once in the Algoritma dan Pemrograman course name

This section is a continuation of the test results that have been described in section 3.1 with input data as shown in figure 2 and table 2. Next, by working on condition entries 2 in table 2, the ALMF system will carry out the tracking process of level 2 directories in level 1. The results of the tracking process as shown in figure 3

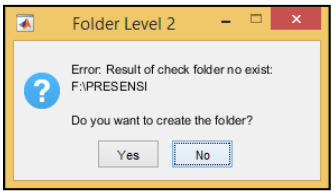

Figure 3. Results of the system tracking process at level 2.

Figure 3 explains that after the ALMF system performs the directory tracking process with the name PRESEN11si as level 2 in level 1, the ALMF system obtains information stating that the PRESENT 11 directory name is not in level 1. The information provided by the ALMF system is not wrong.

To prove that the information is not wrong can be done using two ways, first through file explorer and second through DOS. The results obtained using the second method as shown in figure 4a

```
Folder PATH listing for volume Drive
Volume serial number is AEC5-190F
F:.
No subfolders exist
```

(a)

```
Folder PATH listing for volume Drive
Volume serial number is AEC5-190F
F:.
└───PRESENSI
```

(b)

Figure 4. (a) before and (b) after

Because as level 2 with the name PRESENSI is not yet in level 1, the ALMF system needs to carry out the task of creating a directory named PRESENSI. The command for the ALMF system to carry out the task of creating a level 2 directory with the name PRESENSI is by pressing the "Yes" button in the Folder level 2 dialog box.

To see that the ALMF system has not or has already created a directory with the name PRESENSI can be done using two ways, first through file explorer and second through DOS. The results obtained using the second method as shown in figure 4b.

After the ALMF system carries out the task of creating a directory with the name PRESENSI, then the system performs the directory tracking process with the course name of the Algoritma dan Pemrograman at level 3. The results of the tracking process provided by the system as shown in figure 5.

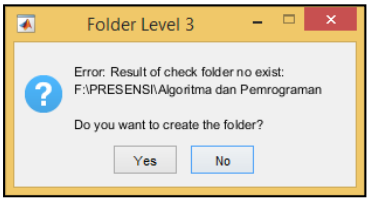

Figure 5. The results of the course name tracking process do not yet exist

Figure 5 explains that the results of the ALMF system tracking process provide information if the directory name of the name with the name Algoritma dan Pemrograman as level 3 directory is not yet in level 2. The information provided by the ALMF system is not wrong because the ALMF system has not received an order to make it. PRESENSI is a directory created by the ALMF system (see figure 4b).

The command for the ALMF system to do the task of creating a level 3 directory with the name Algoritma dan Pemrograman is by pressing the "Yes" button in the Folder level 3 dialog box.

To see that the ALMF system has not or has already created a directory with the name Algoritma dan Pemrograman can be done using two ways, first through file explorer and second through DOS. The results obtained using the second method as shown in figure 6.

```
Folder PATH listing for volume Drive
Volume serial number is AEC5-190F
F:.
└───PRESENSI
    └───Algoritma dan Pemrograman
```

Figure 6. Directory of Algoritma dan Pemrograman

After the ALMF system creates a level 3 directory, then the ALMF system carries out the re-tracking process. In this tracking process to get information about the existence of the college meeting directory as a level 4 directory. The results of the level 4 directory tracking process as shown in figure 7.

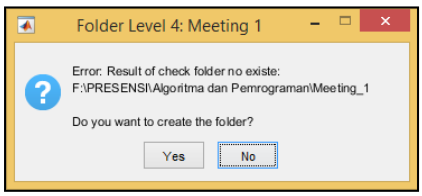

Figure 7. Results of tracking the course meeting folder

Figure 7 explains that the results of the ALMF system tracking process provide information stating the name of the lecture meeting as level 4 directory is not yet in level 3. The command for the ALMF system to do the task of creating a level 4 directory with the name Meeting_1 is by pressing the "Yes" button in the Folder level 4 dialog box.

To see that the ALMF system has not or has already created a folder named Meeting_1 can be done using two ways, first through file explorer and second through DOS. The results obtained using the second method as shown in figure 8.

```
Folder PATH listing for volume Drive
Volume serial number is AEC5-190F
F:.
└───PRESENSI
    └───Algoritma dan Pemrograman
        └───Meeting_1
```

Figure 8. ALMF system is run once.

Figure 8 explains that the ALMF system that is run once for the chosen subject name is Algoritma dan Pemrograman can create a lecture meeting folder for the name of the course.

The results of the ALMF system testing are run once, indicating that for the actual ALMF system, displaying information in a step-by-step dialog box is not needed. The process of displaying any information that is done by the ALMF system can make the execution time longer.

Based on the results of the test it can be concluded that only using data input the drive name, default name, and course name; actually the ALMF system has been able to create an automatic lecture meeting folder and the output generated by the ALMF system is as expected.

3.2 Testing the ALMF system is run twice in the Algoritma dan Pemrograman course name

Based on the results of the test it can be concluded that only using data input the drive name, default name, and course name; actually the ALMF system has been able to create an automatic lecture meeting folder and the output generated by the ALMF system is as expected.

After all input data is approved, the ALMF system will carry out the level 2 directory tracking process with the name PRESENSI not present or already in the level 1 directory. The results of the tracking process as shown in figure 9.

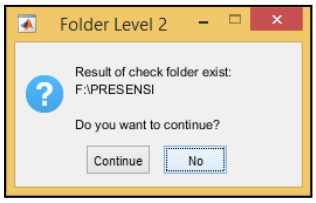

Figure 9. Results of the level 2 directory tracking process

Figure 9 explained that the ALMF system in tracking the level 2 directory in level 1 directory found the default PRESEN11SI name. Information on the results of the tracking process submitted by the ALMF system is correct. To be sure it can be seen again in the results of the hierarchical directory discussed in the previous section.

The command for the ALMF system to do the next process is by pressing the "Continue" button in the Folder level 2 dialog box. The results of the ALMF system tracking process against the level 3 directory as shown in figure 10

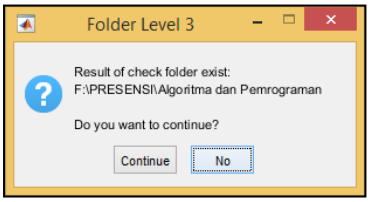

Figure 10. Results of the level 3 directory tracking process

Figure 10 explains that the ALMF system in carrying out the course name tracking process with the name Algo_dan_Pemrog as the level 3 directory has been found in the level 2 directory. The results of the tracking process submitted by the ALMF system are correct. To be sure it can be seen again in the results of the hierarchical directory discussed in the previous section.

The command for the ALMF system to do the next process is by pressing the "Continue" button in the Folder level 3 dialog box. As a result of pressing the "Continue" button, the ALMF system carries out the process of creating a folder named Meeting_2.

To see that the ALMF system hasn't or has already created a folder named Meeting_2 can be done using two ways, first via file explorer and second via DOS. The results obtained using the second method as shown in Figure 11.

```
Folder PATH listing for volume Drive
Volume serial number is AEC5-190F
F:.
└───PRESENSI
    └───Algoritma dan Pemrograman
        ├───Meeting_1
        └───Meeting_2
```

Figure 11. ALMF system is run twice

Figure 11 is an output generated by the ALMF system which is run twice in the course name with the name Algoritma dan Pemrograman.

The results shown by the ALMF system are as expected.

3.4 Testing the ALMF system for N course names and times run

This section describes the implementation of the ALMF system tested in each course name in the course list that corresponds to the input data as shown in the dialog box named Entry level 3 (figure 1c).

In this test, each course name as input data will be used to run the ALMF system as much as M times the course name, where M is a number.

$$M = 1, 2, 3. . ., n$$

The test data used in this experiment for each course name to run the ALMF system as shown in table 3.

Table 3. Data Testing the ALMF System

N	Course Name	Number of ALMF Systems Run (M)
1	Algoritma dan Pemrograman	4
2	Grafika Komputer	2
3	Pengolahan Citra Digital	4
4	Kecerdasan Buatan	3

Table 3 shows the input data for course names with names sequentially Algoritma dan Pemrograman, Grafika Komputer, Pengolahan Citra Digital, and Kecerdasan Buatan used as many as four, two, three, and four times to run the ALMF system.

The results of testing the ALMF system for the test data shown in Table 3 produces outputs called automatic lecture meeting folders that are displayed using the second method, namely DOS as shown in Figure 12.

```
Folder PATH listing for volume Drive
Volume serial number is AEC5-190F
F:.
└───PRESENSI
    ├───Algoritma dan Pemrograman
    │   ├───Meeting_1
    │   ├───Meeting_2
    │   ├───Meeting_3
    │   └───Meeting_4
    ├───Grafika Komputer
    │   ├───Meeting_1
    │   └───Meeting_2
    ├───Kecerdasan Buatan
    │   ├───Meeting_1
    │   ├───Meeting_2
    │   └───Meeting_3
    └───Pengolahan Citra Digital
        ├───Meeting_1
        ├───Meeting_2
        ├───Meeting_3
        └───Meeting_4
```

Figure 12. Lecture meeting folder

Based on the results of the testing on the ALMF system, it gets an analogy between the presence based on the signature with the ALMF system. The analogy findings are as in table 4

Table 4. Presence analogy based on signatures and lecture meeting folders

No.	Signature-Based Presence	Lecture Meeting Folder
1	Signature	Meeting_ folder
2	Number of signatures	Numbers in folders

Comparison of abuse between attendance based on signatures with the ALMF system is in table 5 and table 6

Table 5. Abuse in Signature-Based Presence

No.	Signature-Based Presence	Result
1	Signature	Easy to manipulate
2	Number of signatures	Easy to manipulate

Table 6. Abuse in Lecture Meeting Folder

No.	Lecture Meeting Folder	Result
1	Meeting_ folder	Difficult to manipulate
2	Numbers in folders	Difficult to manipulate

By paying attention to the abuse comparison (table 5 and table 6), the automatic lecture meeting folder system can be an alternative to solving problems that occur in the presence based on the signature.

4. Conclusion

We have presented an automatic lecture meeting folder (ALMF) algorithm based on hierarchical directories. When the expectation of presence at a lecture meeting based on signatures can work according to the rules that apply without manipulation is very small. The existence of this ALMF algorithm has provided a more natural way to solve problems in the presence of lecture meetings based on signatures. The natural way presented by the ALMF algorithm is that every time a teacher runs an ALMF system, the ALMF system will automatically create a lecture meeting folder. This automatically created lecture meeting folder is a form that states the frequency of lecture meetings held by a teacher. Thus manipulation or lack of attention to the presence of lecture meetings based on signatures has been overcome by this ALMF algorithm.

The ALMF algorithm has provided an automatic lecture meeting folder. However, the automatic lecture meeting folder that has been created by the ALMF system with the contents still empty or not available. Therefore, this system still needs further development. For future experiments, the empty lecture meeting folder will contain the face image of the course participants. The use of this facial image is to solve the problem of manipulation that might occur in the presence of lecture meetings based on the signature in the signature column of the lecture participants.

References

[1] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, James W. O'Toole, Jr. Semantic File Systems. SOSP '91 Proceedings of the thirteenth ACM symposium on Operating systems principles. Pages 16-25

[2] Zhichen Xu, Magnus Karlsson, Chunqiang Tang, Christos Karamanolis. Semantic File System. Patent number US7617250B2 (Patent). 2009-11-10.

[3] Y Hua, H Jiang, Y Zhu, D Feng, L Tian. SmartStore A New Metadata Organization Paradigm with Semantic-Awareness for Next-Generation File Systems. SC '09 Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis. 2009. Article No. 10.

[4] Alan C. Lloyd, Lilydala (AU): Susan, M. Oliver, Eaglemont (AU). Directory Sistem. Patent number: US 7.620,630 B2 (Patent), Nov. 17, 2009.

[5] D Davidov, E Gabrilovich, S Markovitch. Parameterized Generation of Labeled Datasets for Text Categorization Based on a Hierarchical Directory. SIGIR '04 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 2004. Pages 250-257.

[6] Margo Seltzer, Nicholas Murphy. Hierarchical File Systems are Dead. HotOS, 2009 – usenix.org

[7] Fredrick Kelley. Interactive knowledge base system. Patent Number: US20030154085A1 (Patent). 2003-08-14

[8] T Fujii, Y Tabata, N Oku, Y Shishido. Automatic storage and retrieval apparatus for individual file folders. US Patent 4,219,296, 1980 - Google Patents.

[9] Nidhra, Srinivas, and Dondeti, Jagruthi, 2012, Blackbox and Whitebox Testing Techniques - A Literature Review, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012

[10] M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). Jurnal Ilmiah Teknologi Informasi Terapan Volume I, No 3, 10 Agustus 2015
[11] Untoro, Wisnu Yudo, F.X. Algoritma & Pemrograman dengan Bahasa Java. Yogyakarta: Graha Ilmu. 2010: 12-13.