Volume 4, Issue 2, February-2017, pp. 53-56 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Analysis of Computational Time on DREAD Model

Didit Suprihanto[#], Retantyo Wardoyo[*]

[#]Doctoral Program of Computer Science at Department of Computer Sciences & Electronics,Universitas Gadjah Mada
Sekip Utara, Bulaksumur, Yogyakarta 55281 Indonesia
[#]Department of Electrical Engineering,Universitas Mulawarman, Kalimantan Timur, Indonesia, 75123
E-mail: didit.suprihanto@ft.unmul.ac.id

*Second Institution, Address, City, ZIP Code, Country
*Department of Computer Sciences & Electronics,Universitas Gadjah Mada
Sekip Utara, Bulaksumur, Yogyakarta 55281 Indonesia
E-mail: rw@ugm.ac.id

Abstract: - In identifying the risks, there are several factors needed to consider, such as the extent to which these risks are exploited and how much damage will occur. Considerations for choosing the most appropriate risk reduction that is fast and safe to perform a good calculation. Calculation complexity is one thing that should be considered in selecting an algorithm to be applied to the decision support system. This paper uses DREAD model by discussing the complexity testing and implement DREAD model into a program. Complexity is used to find out the computation time and its ratio completed with the result that the computation time of the final data is affected by the data addition. Therefore, the addition of data greatly affects to the computation time which is required the ratio of computing time, even though it has a bunch of similar data computation time and in fact these have different results that the ratio of computation time does not give any effect (stable). Computation ratio changes from the initial data group until the end of data group are not significantly compared with the value of computing time for each additional 100 tested data.

Keywords— Complexity, Computing, DREAD

---

1. Introduction

In identifying the risks, there are several factors to consider. These considerations include the extent to which these risks are exploited and how much damage will occur. The aim of this consideration is for choosing the most appropriate risk reduction, fast and secures [1]. There are many methods and models to solve the problems and assessment of the risk on the web application. Some of the methods and tools that can be used to assess the risk are NIST (*National Institute of Standard & Technology*), FRAP (*The Facilitated Risk Assessment Process*), COBRA (*The Consultative Objective and Bi-functional Risk Analysis*), *DREAD Model*, OCTAVE (*Operationally Critical, Threat, Asset and Vulnerability Evaluation*) and *Risk Watch* [2].

In line with the opinion above, it requires a countermeasure risk which is fast, accurate, and safe to make a good calculation. Complexity calculation is one thing that should be considered in selecting an

algorithm to be applied to the decision support system [3].

The complexity is a part of computer science that is related to the use of algorithms in certain applications. The complexity of an algorithm is a step procedure which is used for analysis, such as automated data analysis, processing, and reasoning. Algorithms can be classified based on the amount of time required to complete that compared to the size of the input. There are various varieties such as some complete algorithms in linear time, relative to the size of the input; Another complete amount of time as exponential or worse; and while others are stopped[4].

The complexity of an algorithm is divided into two, they are the complexity of time and space complexity. The time complexity is measured by the number of stages of computation required to run the algorithm as a function of a number of data n (input size). While the space complexity is measured by memory used on the structure of data contained in the algorithm as a function of a number of data n [5].Another opinion about the complexity is intended to test and find out the time, memory and other resources that are needed by algorithm to solve computational problems [6].

In this paper, the risk calculation model is using DREAD model. DREAD model is a model from Microsoft that is used to calculate the risk which can generate information about risk rating for threat occurs. Elaboration of DREAD model is a question that is answered by the respondents consisting *Damage potential, Reproducibility, Exploitability, Affected user, Discoverability* [7].

This paper discusses the complexity of testing, which implement DREAD model into a program. Complexity referred in this discussion is only focused to find out the computation time and the ratio of computation time.

The discussion of this paper is divided into four sections. The first section is an introduction that describes the background of the problem as well as a review of several studies related to the research conducted. The second section describes materials and methods, the section three is the result and discussion, and the section four is the last part and it is closed with a conclusion.

2. The Material And Method

This experiment uses a computer with the following specifications:

- Intel (R) Core (TM) 2 Duo CPU T8300 2.40
- RAM, 1 GB
- Operating system Windows 7 Ultimate 32 bit
- Matlab R2011b

The test data is obtained from the result of questionnaire DREAD model with many respondents 1450. The experiments test is performed with the count data and starts from 50, 100, 150, 200, ... etc. (difference 50) until 1450.

The procedure of testing:

- Prepare the test data which is obtained from the result of questionnaire DREAD model in the amount of 1450 the number of respondents.
- Prepare a program that has been made with Matlab R2011b for a test of computing time.
- Take the test data starts from 50.
- Testing of computing time is performed on each additional of 100 test data, starting from 50, 150, 250, ... until 1450.
- The tests are conducted as much as 30 times repetitions on each group of test data.
- Record the test results of computing time for each group of test data.
- Calculate the average of computation time for each test data groups of the test results (taking the average computation time by looking at the standard deviation) and it can be seen in Table 1.
- Calculate the average ratio = pd group / the largest average
- Create a graph and analysis

3. Result and Discussion

This section discusses the implementation of each data into the program that was created to test the computing time ,which is owned by each group of data. Testing time computation is performed on each additional 100 test data, starting from 50, 150, 250, ... up to 1450. Each test was performed 30 repetitions on each

group of test data by finding the average value for each group of data. Measurement of computing time that is performed 30 times to ensure accuracy of computing time, which is owned by each group of data in addressing a number of different data input. The following section is a table of result tests that have been done. Time measurement is done by units of microseconds. Summary of test results on DREAD Model computing time is shown in Table I.

TABLE I Result Of Computation Time

NO	Number of testing (NT)	Computation Time (microsecond) (CT)	RATIO (CT/NT)
1	50	24	48%
2	150	57	38%
3	250	90	36%
4	350	118	34%
5	450	153	34%
6	550	183	33%
7	650	215	33%
8	750	244	33%
9	850	277	33%
10	950	315	33%
11	1050	340	32%
12	1150	362	31%
13	1250	404	32%
14	1350	436	32%
15	1450	474	33%

From the Table 1 above, it is known that there are many data that are tested with the difference between 100 and starts from 50 to 1450. The test results of computation show the result of average computation time starting from 50 to 100, until the count data is 1450. From these results, it can be known that the computing time results are increased from each group of test data. The increase in computing time is caused by the addition of test data which is performed. However, the addition of test data that is fixed / constant is not equal to the test results of the computation time of each group. Graphic Computation Time can be seen in Figure 1.

Figure 1 shows that the computing time of the initial data group until the end of the data group and having a linear graph, the graph, equation Y = 0.316X + 8892, this means that if additional data (X) as many as 100 data, it will add to the computing time of 31.6 microseconds, so extra data or affecting the computation time required.

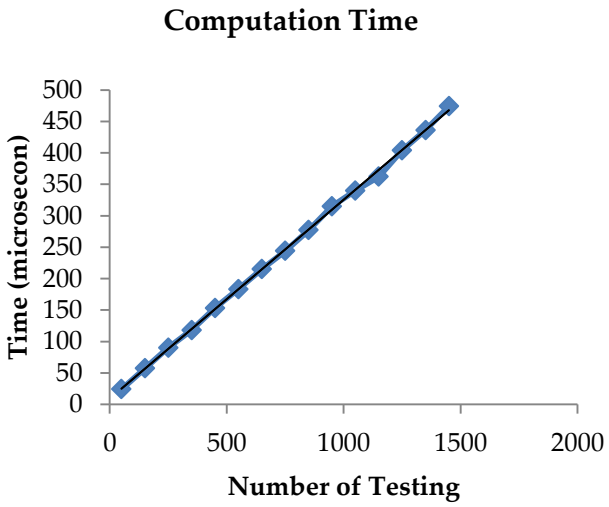

FIG. 1 RESULT OF COMPUTATION TIME

While the ratio in each group tests data that is shown in Table 1 can be seen in Figure 2.

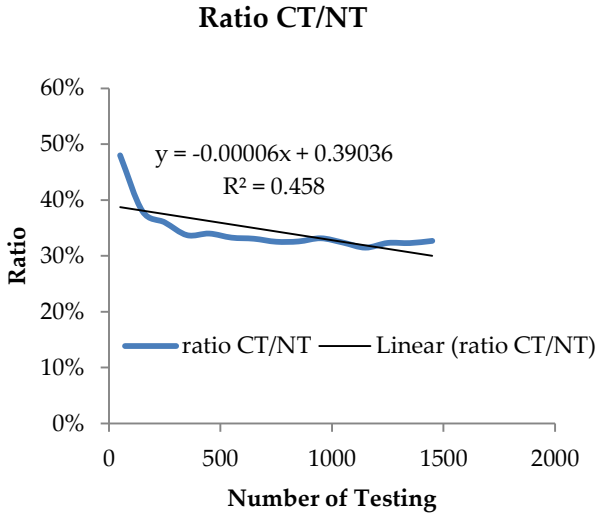

Fig. 2 Ratio of computation time testing

Figure 2 shows that the ratio of computing from the beginning data group to the end of the data group has a linear graph, with the graph ,equation Y = -0.00006X + 0.39036, this means that if additional data (X) as many

as 100 data will affect or reduce the computing time ratio of 0006, or by 0.6%. Therefore, it can be said that the addition does not affect or remain at the ratio of computation time..

4. Conclusion

Based on the results of tests that have been performed, it can be seen that in general the computing time and the ratio of computation time with the experimental group of the same data, it proves that based on the value of computational time gain values rise. Computing time value is in line with the addition of data. The addition of the value of the computing time follows a linear trend equation rise (positive), and it can be interpreted that the computation time of the initial data group until the end of the data groups affected by the addition of data, so the addition of the data greatly affects to the computation time required.

However, the ratio of computation time, despite having a group of the same data with computational time, in fact have different results, where the ratio of computation time can be said to be no effect or fixed (stable). Computation ratio changes from the initial data group until the end of the data groups was not significant compared with the value of computing time for each additional 100 tested data.

References

[1] McEvoy, N., Whitcombe, A.,. Structured Risk Analysis. *International conference on infrastructure security,* vol. 2437, Bristol , October 1-3, 88-103. 2002

[2] Elky, S., An Introduction to Information System Risk Management. SANS Institute InfoSec Reading Room.copyright©SANS Institute. 2006.

[3] Hamdani., Wardoyo, R., The Complexity Calculation for Group Decision Making Using TOPSIS Algorithm. Advances of Science and Technology for Society AIP Conf. Proc. 1755, 070007-1–070007-7; doi: 10.1063/1.4958502 Published by AIP Publishing. 978-0-7354-1413-6: 2016.

[4] M. Ölmez and U. Lindemann, Procedia Comput. Sci. 28, 130 :2014.

[5] I. Wegener, Complexity Theory (Exploring the Limits of Efficient Algorithms) (Springer-Verlag, Dortmund), pp. 1–380: 2005

[6] Sipser, Michael, Introduction to the Theory of Computation – Second Edition, Thomson Course Technology, Massachusetts : 2006.

[7] Meier, J.D., Mackman, A., Vasireddy, S., Dunner,M., Escamilla, S., Murukan, A.,. Improving web application security: Threats and Countermeasures. Microsoft Corporation.2003..

