

Cite as: Ali, A., & Wibowo, K. (2016). Using office simulation software in teaching computer literacy using three sets of teaching/learning activities. *Issues in Informing Science and Information Technology*, 13, 79-88. Retrieved from <http://www.informingscience.org/Publications/3470>

# Using Office Simulation Software in Teaching Computer Literacy Using Three Sets of Teaching/Learning Activities

**Azad Ali and Kustim Wibowo**  
**Indiana University of Pennsylvania, Indiana, PA, USA**

[Azad.ali@iup.edu](mailto:Azad.ali@iup.edu) [Kustim.wibowo@iup.edu](mailto:Kustim.wibowo@iup.edu)

## Abstract

The most common course delivery model is based on teacher (knowledge provider) - student (knowledge receiver) relationship. The most visible symptom of this situation is over-reliance on textbook's tutorials. This traditional model of delivery reduces teacher flexibility, causes lack of interest among students, and often makes classes boring. Especially this is visible when teaching Computer Literacy courses. Instead, authors of this paper suggest a new active model which is based on MS Office simulation. The proposed model was discussed within the framework of three activities: guided software simulation, instructor-led activities, and self-directed learning activities. The model proposed in the paper of active teaching based on software simulation was proven as more effective than traditional.

**Keywords:** Simulation Software, Office Simulation Software, eBook and Simulation Software

## Introduction

Information Technology (IT) educators have long faced criticism for the selection of content they cover, and for their methodology in which they deliver the information (Daempfle, 2003, Gallivan, Truex and Kvansy, 2002, Radermacher, & Walia, 2013). The author of this paper had his share of criticisms for his reliance on textbooks and tutorial solutions provided by the publishers of the textbooks. It is no secret that many IT textbooks provide tutorials and step-by-step instructions that guide the students to work through exercises. This use of tutorials is not limited to one area of IT teaching. Textbooks with these kinds of tutorials and are widely used at IT courses and can be found in all areas of IT such as computer literacy (Grauer et al., 2010, Poatsy et al., 2013), programming languages (Bradley & Millspaugh, 2003; Zak, 2015), web design (Bishop, Shuman, & Waxer, 2009; Carey, 2009), networking (Hallberg, 2009; Tomsho, 2015), database management (Friedrichsen, 2015; Shelly, Cashman, Pratt, & Last, 2007).

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

Technological advances took the textbook tutorials a step further. Publishers now can include in their textbooks simulation steps for the students to follow and complete their exercises. This helped educators to use the simulation software in their classrooms which in turn can help to make teaching clearer; they simplify the delivery of knowledge and they may at the end contribute to more effective teaching/learning (Fen-

**Editor: Eli Cohen**

Submitted: November 30, 2016; Revised: February 29, 2016; Accepted: March 1, 2016

wick, et al. 2013; Hill, 2011). However, the complete reliance on these technologies may be detrimental to teaching and may prove to be counterproductive as well. By solely using these technologies, it will diminish the role of the teacher and may transfer the full responsibility for the teaching to students themselves by their reliance on the textbook and the software technology.

Taking both points of view that were explained in the previous paragraph, a balanced approach to using these technologies in the classroom may be deemed more helpful. This balanced approach encourages the students using the simulation software and also boosts a bigger role to the teaching of the faculty. This policy is followed by the authors of this study and they are illustrating their experience in this paper. The remainder of this paper is divided into the following sections:

- First, the paper provides a literature review about software simulation, their origin, features and their use in academic courses
- Second, the paper continues to provide literature review about three set of teaching activities, describing each one of it and explaining their effectiveness and weaknesses.
- Third, the paper illustrates the experience of the authors that used the three set of activities along with the simulation software to teaching computer literacy courses
- Last, the paper summarizes the content of this study and suggest additional research for this topic.

## **Literature Review - Office Simulation Software**

Although office simulation software has been in use by educators for some time, their origin, their features, and their other activities remain unknown to some. Thus this section sheds light on the use of simulation software in general and focuses more on Microsoft Office simulation in more depth. It begins by explaining the meaning of the word simulation and continues to discuss computer software simulation, the features common in Office simulation software, and the automatic grader in Microsoft Office software simulation.

### ***Simulation – Meaning and Brief History***

Bell and O’Keefe (1987) noted that simulation has existed for many years. They pointed to 1965 as a starting year and gave an example about the simulation from a steel melting shop to show simulation experiments. However, before delving more into the history and uses of simulation, we thought that understanding the meaning of the word “simulation” may help to learn more about their use in academic activities.

Our quest to find the meaning of the word “simulation: landed first with the online dictionary. Webster’s dictionary for example, defined simulation as “the imitative representation of the functioning of one system or process by means of the functioning of another” (Simulation, 2015). While this definition seems simple enough and does not warrant more explanation, simulation can be looked at from a different perspective and could be understood differently. Maria (1997) for example looked at simulation from the operation prospect and defined it in the following:

A simulation of a system is the operation of a model of the system. The model can be reconfigured and experimented with; usually, this is impossible, too expensive or impractical to do in the system it represents. The operation of the model can be studied, and hence, properties concerning the behavior of the actual system or its subsystem can be inferred. In its broadest sense, simulation is a tool to evaluate the performance of a system, existing or proposed, under different configurations of interest and over long periods of real time. (p. 7)

Carson and John (2004) combined simulation with models and distinguished simulation model from discrete model “A model is a representation of a system or process. A simulation model is a representation that incorporates time and the changes that occur over time. A discrete model is one that changes only at discrete points in time, not continuously” (p. 9).

## ***Computer Software Simulation***

Computer software simulation can mean one of two things:

- It can mean the computer software that is written to facilitate the simulation process (simulation software)
- It can also mean the simulation software or program that is written to explain (simulate) the work of other programs and other software applications

In this study, we are interested in the later meaning of computer software simulation. We are interested in the simulation software that facilitates the learning of other software. We believe that two developments led to expanding the use of software simulation:

- Advances in technology made it easier to include many animation features in a document. It is easier nowadays to include moving objects, pointing substances or any other form of animation to simplify the explanation of functioning software
- The software application themselves have grown in complexity and they need to be explained using more visual objects.

It is no longer sufficient to explain the functioning of a software application by simply listing the steps in a text document. The inclusion of visual objects makes the presentation more clear; it makes it more interesting and helps attracting the interest of the students. Thus, the inclusion of software simulation increased. At the same time, the functionality of computer software increased in sophistication; their features increased in their working and the options that they offer increased as well.

## ***Microsoft Office Simulation - Features***

Microsoft Office simulation is not a new brand of simulation. Instead, it is a continuation of older computer software simulation. Microsoft has long developed tutorials that guide users and explain to them the functionality of their software (Pratap, 2009). This was included in the form of text, video, animation, simulation, and many other forms. However, what is more recent in this regard is the use of simulation software by textbook publishers. Hill (2011) highlighted the following functions available in the office simulation software that are provided by publisher of computer textbooks:

- They offer a variety of assessment tools that include test banks, simulators, and document checkers
- They work by first assigning tasks to students and then record the student’s keystrokes as they work on the assignments
- At the end, the software compares the students submitted work with possible correct actions and the results are compared (graded) accordingly

Hill (2011) noted three of these office simulations that are commonly used by textbook publishers: SAM by Cengage learning, MyITLab by Person Education, and SimNet by McGraw Hill higher education. While the software by the three publishers may differ in the details of their work, they have some common features. Among the common features include the following list of activates that they provide for the courses that use them:

## Using Office Simulation Software

- Software guided activities
- Individual simple case activities
- Comprehensive case projects
- Testbank activities
- Inclusion of outside projects

In this study, we are interested in the use of simulation software for teaching four of the Microsoft office suite of applications: Microsoft Word, Microsoft Excel, Microsoft PowerPoint, and Microsoft Access.

### **Office Simulation – Automatic Grader**

Within the office simulation software provided by the publishers is often included a functioning system that automatically grades the activities submitted to it - called the automatic grader. The automatic grader for Microsoft office files can best be explained in terms of submitting a Word document to it. Hill (2011) explained the process by which a Word document gets graded when it is submitted to it in the following:

- When a document is submitted to the grader, the automatic grader then uses Microsoft software named “compare and combined” to compare the submitted document with the correct version of the document.
- The grader then counts the errors and grades the submitted document accordingly

The grading process described above for the Word document is similarly followed in all other Microsoft office documents. The interesting feature of the office grader is the immediate grade and feedback that it provides to the students taking the course and the faculty teaching it. Since all this is automated and done through the computer, it is quicker than manual grading and it provides assignment notes that often exceed the notes provided by a reviewing faculty. The notes include pointer to the exact location of errors, suggestions for correction, points taken off and reasons for taking the points, statistics about time, word count, and many other data. Thus if a faculty uses this automatic grader, he/she will have sufficient data to examine the assignment and suggested solutions for the problems in the submitted assignment.

## **Literature Review - Three Teaching Activities**

This section reviews literature about three different activities that are included in the teaching of the two authors for their computer literacy courses. These set of activities include using guided software simulation (termed as tutorials), using teacher guided activities, and then having students self-directed to work on their own to complete the activities.

### **Using Guided Software Simulation as a Startup**

The use of guided steps as dictated by computer software is often practiced in tutorials, software installations, and other uses as well. This approach asks the users to perform certain tasks and waits for the user entry. After the user enters or performs keystrokes, the software assess the correctness of the entry. If the entries are correct, it moves on to the next step. However, if the entries are not correct, the simulation software makes the correction by showing a demo of the steps need to be completed, or points to the location where the demo can be reviewed.

Dong et al. (2012) noted the following about using tutorials for learning: “Tutorials are a popular form of help content, as they offer step-by-step instructions for how to accomplish a task. But step-by-step demonstrations do not encourage deep understanding and do not facilitate transfer of skills to new contexts” (p. 2083). Brown (2001), on the other hand, called this type of activity

“computer based learning” and contrasted it with “instructor-led training” and noted that the first offer learners more control over their instruction.

The use of these type of activities in the classroom has advantages; it allows the user to learn directly from the software, thus increasing the reliance on the machine. The interactive nature of these tutorials may also increase attention of the students and simulates their interest in the topic (Bell & O’Keefe, 1987; Fenwick et al. 2013). However, there is little role for the instructor when teaching in the classroom. The role of the instructor may include clarifying issues, troubleshooting problems, and drawing parallels to connect to other topics. All things are explained in the tutorials and there is little room for errors thus reducing the chances for instructor role.

### ***Instructor-Led Learning Activities***

Jones, Mooeni, and Ruby (2005) called this “traditional learning” activities and explained it as activities that take place in the classroom under the direction and guidance of the instructor. While this was the norm in all software training before the expansion of e-learning and computer based-learning, this continues to be a viable option for teaching and learning.

Under this set of activities, the instructor works on the specific software, follows the steps to accomplish certain tasks, then allows the students to follow and complete what was demonstrated to them. There could be intermittent explanations and answering questions for the steps completed and/or concepts explained. This method is aided by the recent advances of classroom technologies which included overhead projector, PowerPoint presentation and other hardware such as a microphone and large screen presenter. In other words, the teacher works on an activity, shows the work on overhead projector, and the students see what the teacher is doing and can follow the steps completed.

Using this set of activities, the instructor assumes a bigger role to play in the learning process. His/her mastery of the software work may help overcoming problems as they occur during the demonstration of the work on the software. These types of activities also may be more suitable for comprehensive projects where questions by the students increase and are more likely to be asked by more than one student. So instead of answering the same question multiple times, using these kind of activities, the instructor can answer it for the entire class. At the same time, using these kind of activities, the students mostly work at the same pace. While this will be ok in most cases, for some quicker students it may delay them or put them behind in completing the activities because they have to wait for other students to complete the steps before moving to the next.

### ***Student Self-Directed Learning Activities***

In self-directed activities, the students are given work (like assignments or exercises) and they are asked to complete them on their own, using their own resources instead of asking the teacher to show them how to complete the exercise. This can be an effective form of learning if sufficient perpetration and guidance are included with it. Abraham, Upadhya, and Ramnarayan (2005) explained that self-directed learning combined with sufficient guidance promotes deeper learning and provides better recall.

Experts often stress the importance of having the students relying on themselves in solving some problems. Without this approach, the student learning remains vague and unaccounted for (Brown, 2001; Jones et al., 2005). However preparation and explanation is necessary before giving the students these tasks (Abraham et al., 2005). Suitable projects for these types of activities may include showing the students one kind of project and assign the students’ somewhat different projects.

Technological advances helped with this kind of activities as well. Students can easily search the web and look for answers as they work on these activities. Nowadays, the learning management system (LSM) can help with these activities. In LSM text/video/chatting/email can be used to help with these activities. The role of the teacher in these kinds of activities is limited if it is not preceded by bigger role (like the inclusion of instructor-led activities before using these activities) of the teacher. This kind of activity is more effective when it is preceded by teaching the foundation by the instructor or from other sources. Otherwise, just giving the students assignments and asking them to research for the solution will not be fair to the students and will not be productive for the learning process.

## **Using Simulation Software in Teaching Computer Literacy Courses**

This section explains the use of simulation software in the teaching of computer literacy courses within the three teaching activities that were discussed earlier in this paper. It illustrates the experience of two faculty members who have taught this course, have applied the simulation technology in their courses, and have used them within the three teaching activities. It first begins by providing background on the course and the selected software tool. It is then followed by explanation of each teaching activity and how the faculty uses the technology to apply the activity in their courses.

### ***The Computer Literacy Course***

The course IFMG101 – Computer Literacy is a three-credit first year course that does not have any pre-requisite. Students can take this course online or in the classroom (what is termed as Face-to-Face). A major portion of the course includes teaching popular productivity software tools using four major software applications: word processing, spreadsheet, presentations, and database management systems. Assignments were given for each of the applications. Naturally, our selection focused on Microsoft Office application: Word, PowerPoint, Excel and Access. The authors used in the past textbooks supplied by the publisher Prentice Hall to teach the application concepts and practices for this course. Lately the authors switched to using eBooks and using the office simulation software tools.

### **The selected office simulation software tool**

The selected simulation software for this is Prentice Hall's MyITLab which is bundled with an eBook by the same publisher. The selected eBook is a continuation from the old textbooks that we have used for this course. Different faculty members at our campus use this eBook for the course. There are different features and different group of activities that led us to the selection of MyITLab for this course, this includes the following features:

- Software guided activities
- Small case projects
- Comprehensive case projects
- Instant grading & feedback

The software guided activities are termed “simulation exercises” software by MyITLab. They guide the students into completing the exercises step by step. The software helps the students to correct and resubmit their assignments. As the students work on exercises, the software can demonstrate the correct steps to complete the given exercise. It then allows the students to redo the work based on given number of times set by the teaching faculty. It explains the source of the error and the recommended corrected steps. The corrected steps can be shown in either a video

clip or it can be shown through the demonstration of required steps. The corrected steps can also be viewed in text messages.

The small case projects are called “grader activities” by MyITLab. They are composed of instructions to complete exercises for practical business applications. The instructions in this case are stated in words and sentences rather than visually as in the simulation exercises. Data files are provided to start each activity and instructions can be downloaded to complete the activities.

The Comprehensive case projects are called as “capstone projects” in MyITLab. They contain projects that are longer, more complicated, and require more steps to complete than in the other activities. Completing these exercises requires more knowledge and sophistication than required in the simulation exercises and the grader activities. Basically, completing a capstone project may practically mean going over all the points that need to be covered in the chapter in a practical, widely used application.

The instant grading and feedback is available in all three activities in MyITLab. Once the students complete the activity, they can submit it for grading and feedback. The grade for their work is shown by the software in MyITLab shortly after the submission. Detailed feedback is returned back from the grader as well. The feedback points to the location of the error, and it explains the error and provides room for corrections and resubmission.

### **Teaching with software simulation activities**

In this group of activities, we assign the students to work on the simulation exercises. We assign these activities at the beginning coverage of the chapter. We begin by explaining the concepts that are covered, give an overall snapshot of the steps that are to be completed in these simulation exercises, and then caution the students about any problematic areas that they may encounter as they work on the exercises. Then we let the students work on the exercises. We give the students a maximum of three attempts to submit their completed work to the grader. Our role for this type of activities is limited to the times when the students have problems or face difficulties in completing the exercise. In these cases, we either explain to the students how to complete it or show them where they can search and find the information to solve the problem.

### **Teaching with instructor-led activities**

We use the “Instructor-Led Learning Activities” to direct the students to work on the capstone projects. As explained earlier, capstone projects are comprehensive case studies that applies all concepts covered in the chapter. They are long projects that require combining multiple skills, competencies, and concepts that are covered in the chapter. Thus we work with the students to explain the concepts in detail, to show them how to complete each step, and then to guide the class on completing these capstone projects.

We begin each capstone project by explaining the concepts that are covered in it. Then we show them the beginning of steps as they download the required files from the publisher web site. This typically is followed by showing to the students the final version of the assignment as it supposed to look when they complete it. Last we explain the steps that are followed to complete the capstone exercises.

After giving all the necessary background information to work on the capstone exercise, we start working on the exercises and guide them to complete and submit them for grading. This happens by working on the exercises ourselves step by step and showing the students on a large projector screen. As we complete each step, we explain to the students what was completed and ask them to follow the steps. We also take the time to help the students when they have problems with completing the steps so all of them complete the capstone project at the same time.

Once we complete guiding the students to complete the capstone activity, we ask them to submit it for grading. The students submit their work for grading and can read the feedback provided by the software and try to correct their errors and submit it again. We allow three submissions for the capstone exercises. If the students still have trouble after submitting it second time, they ask us and we help them. However, we ask the students to go through this step (submitting and reading their errors) at least once. We feel that having the students go through reading the error messages and attempting to fix the errors will help the students have a better grasp of the materials covered.

### **Teaching with student self-directed learning activities**

In this group of activities, we assign the students to work on the grader exercises. Recall from earlier discussion that grader activities are cases smaller than the capstone exercises; they contain instructions on steps to complete the case but their instructions are not visual as the case in the simulation exercises. In terms of their difficulty level, grader activities are placed somewhere between the simulation exercises and the capstone projects.

Our role in this set of activities is to respond to questions from the students and point them also to where they can research and find the answers. We encourage the students to work on their own to find solutions for the issues raised as they work on the grader activities. We feel that having the students go through all of the steps required to complete the grader activity (this includes completing the steps, submitting their work for grading, reading notes from the grading, and research for answers) will give the students more confidence in what they learned. Thus we stress on the students to complete the grader activities on their own and we respond to questions only after the students do their work researching the solution.

### **Summary and Suggestion for a Future Study**

This study is about using MS Office simulation software in teaching computer literacy courses. The paper began by explaining about simulation software and gave a literature review about their origin, their features and then specifically about office simulation. It then discussed three sets of activities that are typically incorporated in the teaching of software applications. It reviewed literature about each activity sets, it explained their common use and their points of effectiveness and weaknesses for each. The paper then shifted to talk about the experience of the authors of this paper and their use of the simulation software in their teaching of computer literacy courses combined with the three sets of activities.

The use of the simulation software in the teaching of our computer literacy course was helpful in many ways, but there was another factor that helped in this regard as well. In our opinion, the use of eBooks in this course helped solved many of the problems that were generated from using regular textbooks in previous courses. This in turn helped with the effectiveness of using the simulation software in our course. Thus we intend to pursue another study regarding the use of eBooks in computer literacy courses and we will focus on our experience of requiring them in our courses.

### **References**

- Abraham, R. R., Upadhyaya, S., & Ramnarayan, K. (2005). Self-directed learning. *Advances in Physiology Education, 29*(2), 135-136.
- Bishop, S., Shuman, J., & Waxer, B. (2009). *The WEB collection revealed standard edition: Adobe Dreamweaver CS4, Adobe Flash CS4, and Adobe Fireworks CS4*. Cengage Learning.
- Bell, P. C., & O'Keefe, R. M. (1987). Visual interactive simulation—history, recent developments, and major issues. *Simulation, 49*(3), 109-116.



- Bradley, J. C., & Millsbaugh, A. C. (2003). *Advanced programming using Visual Basic*. NET. McGraw-Hill/Irwin.
- Brown, K. G. (2001). Using computers to deliver training: Which employees learn and why? *Personnel Psychology*, 54(2), 271-296.
- Carey, P. (2009). *New perspectives on creating web pages with HTML, XHTML, and XML*. Cengage Learning.
- Carson, I. I., & John, S. (2004, December). Introduction to modeling and simulation. In *Proceedings of the 36th Conference on Winter Simulation* (pp. 9-16). Winter Simulation Conference.
- Daempfle, P. A. (2003). An analysis of the high attrition rates among first year college science, math, and engineering majors. *Journal of College Student Retention: Research, Theory & Practice*, 5(1), 37-52.
- Dong, T., Dontcheva, M., Joseph, D., Karahalios, K., Newman, M., & Ackerman, M. (2012, May). Discovery-based games for learning software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2083-2086). ACM.
- Fenwick, J. B., Jr., Kurtz, B. L., Meznar, P., Phillips, R., & Weidner, A. (2013, March). Developing a highly interactive ebook for CS instruction. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 135-140). ACM.
- Friedrichsen, L. (2015). *Illustrated course guide: Microsoft Access 2010 Basic* (1st ed.). Cengage Learning.
- Gallivan, M., Truex, D. P., III, & Kvasny, L. (2002, May). An analysis of the changing demand patterns for information technology professionals. In *Proceedings of the 2002 ACM SIGCPR Conference on Computer Personnel Research* (pp. 1-13). ACM.
- Grauer, R. T., Poatsy, M. A., Hulett, M., Krebs, C., Mast, K., Mulbery, K., & Hogan, L. S. (2010). *Exploring Microsoft Office 2010, Volume 1*. Prentice Hall Press.
- Hallberg, B. (2009). *Networking, A beginner's guide*. McGraw-Hill, Inc.
- Hill, T. G. (2011). Word grader and PowerPoint grader. *ACM Inroads*, 2(2), 34-36.
- Jones, K. R., Moeeni, F., & Ruby, P. (2005). Comparing web-based content delivery and instructor-led learning in a telecommunications course. *Journal of Information Systems Education*, 16(3), 265.
- Maria, A. (1997, December). Introduction to modeling and simulation. In *Proceedings of the 29th Conference on Winter Simulation* (pp. 7-13). IEEE Computer Society.
- Pratap, R. (2009). *Getting started with MATLAB: A quick introduction for scientists and engineers*. Oxford University Press, Inc.
- Poatsy, M. A., Krebs, C., Cameron, E., Williams, J., & Grauer, R. (2013). *Exploring: Microsoft Access 2013, Comprehensive*. Prentice Hall Press.
- Radermacher, A., & Walia, G. (2013, March). Gaps between industry expectations and the abilities of graduates. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 525-530). ACM.
- Tomsho, G. (2015). *MCSA Guide to Administering Microsoft Windows Server 2012/R2, Exam 70-411* (1st ed.). Cengage Learning.
- Shelly, G., Cashman, T. J., Pratt, P., & Last, M. (2007). *Microsoft Office Access 2007: Comprehensive concepts and techniques*. Cengage Learning.
- Simulation (2015). In Merriam-Webster.com. Retrieved June 20, 2015 from <http://www.merriam-webster.com/dictionary/disruption>
- Zak, D. (2015). *Programming with Microsoft Visual Basic*. Cengage Learning.

## Biographies



**Azad Ali, D.Sc.**, Professor of Information Technology at Eberly College of Business – Indiana University of Pennsylvania – has 30 years of combined experience in areas of financial and information systems. He holds a bachelor degree in Business Administration from the University of Baghdad, an M.B. A. from Indiana University of Pennsylvania, an M.P.A. from the University of Pittsburgh, and a Doctorate of Science in Communications and Information Systems from Robert Morris University. Dr. Ali’s research interests include service learning projects, web design tools, dealing with isolation in doctoral programs, and curriculum.



**Kustim Wibowo, PhD.**, Professor of Information Technology at Eberly College of Business – Indiana University of Pennsylvania – has 30 years of combined experience in areas of financial and information systems. He holds a bachelor degree in Business Administration from the University of Indonesia, Jakarta, Indonesia, an M.S. in Computer Science from Baylor University of Waco, Texas and a Doctorate of Philosophy in Management Information Systems from University of Kentucky, Lexington, Kentucky. Dr. Wibowo’s research interests include big Data Management Systems, Decision Sciences, and Information Security.