Database Design for Distribution Simulation Game

A Santosa1*, A M Rinaldy2, D Andriani3
1,2,3Industrial Engineering Department, Universitas Komputer Indonesia, Indonesia

Email : *alam.santosa@email.unikom.ac.id

Abstract. The purpose of this study is to obtain a database design for a distribution simulation game that will be used as a learning medium in an industrial engineering. Distribution simulation games have been used since the 1960s, since the invention of the Beer Game by Jay Wright Forrester. The method used in this research is the System Development Life Cycle including the steps of enterprise modeling, conceptual data modeling, logical database design, physical database design, and database implementation. The results of this study are in the form of a relational database design consisting of five entities namely GameTitle, Retailer, Wholesaler, Distributor, and Manufacturer. GameTitle is a strong entity that owns the other four entities. The design can be implemented well in a Database Management System such as Microsoft Access, MySQL, or others. This database is used as data storage during the game and then used as a data source to analyze the effectiveness of understanding the concept of learning in industrial techniques such as forecasting, inventory planning, and supply chain management. The main conclusion that can be drawn in this study is that the database design for the distribution simulation game can be implemented using Microsoft Access with the condition that transaction data between echelon is stored in a separate table to avoid conflicts when accessing simultaneously. With this database design, it is expected that everyone can develop a distribution simulation game application for improving the learning process in their respective fields of study.

1. Introduction

The distribution simulation game is a computerized game which made with supply chain concept. This game describing about inventory manager on several location called echelons. The manager must make a decision as to what level of inventory to decide. A number of demands will occur every period and take the inventory if possible, and if it is not, a total of lack will increase the number of backlogs.

The research about making the game was made by any researcher in the world, especially simulation game for example Simplan, OMG! [1]. The original beer game from MIT made by Forrester in 1960. The beer game by Forrester was made to solve the problem about production fluctuation on household appliance factory in Kentucky. In that time, the game is a physical simulation with paper and pencil, its purpose is to find the bullwhip effect on their inventory management [2]. In 1989, there is a development about beer game. Many researchers were made it with any style and any rules. However, there is no research that explains about the simulation database’s design. This study aims to produce alternative database designs for similar games so that they can be implemented in Database Management System (DBMS) applications such as Microsoft Access. A good database design will be able to generate information about how students respond to system dynamics by applying the theories they learn in Industrial Engineering.

After the database design is obtained, the research can proceed to the development of a distribution simulation game application. A good single database design can be used in many simulation application models [3]. Therefore, it requires accuracy and caution in its development steps. The game
will give experience how to define a number of stock so that every players have a minimum the backlog values or does not occur as much as possible [4][5]. It is expected to be able to provide experiential learning about making forecasting decision and helping player to understand how importance of forecasting [6]. The purpose of this study is to find a database design for a distribution simulation game using the system development life cycle approach (SDLC) [7] as described by Hoffer et al. The SDLC approach used is Agile SDLC which has more communication so as to enable product development in a short time with more flexible documentation [8].

2. Methodology
The method of developing this game is adapting from System Development Life Cycle (SDLC) [9] and some reference by philosophy of Monopologs [10]. First is analyze the system which we do. System analysis made to determine what information is needed to support the object and function of research. Second, system design, which is a system that is used in all types of design and as a whole in which one problem that connected to each other. The design stage displayed data flow diagrams. Third, system implementation, an activity that is planned and carried out seriously also refers to certain norms in order to achieve the objectives of the activity. The results of the implementation phase are implementing the database [11]. A database is defined as a description of the data using notation that matches the object data, generated from a database management system (DBMS), which is software for storing, obtaining, and protecting data. The DBMS used in this study is Microsoft Access. While the application is a product of the results of the DBMS, namely distribution simulation game made with Microsoft Access.

The philosophy of this game is how to make the inventory or stock can fulfill the incoming demand by period. Sometimes the numbers of demands are too high or sometimes are too low of the inventory. Each player must have a calculation and forecasting about this condition. The random demands are generated automatically by system using an LCG calculation from Monte Carlo simulation. Each period, when demands are coming the system will automatically take the inventory for fulfill the number of demands. If the inventory is enough to cover the demands, then the number of inventory will decrease and the deliver tab will increase with the same values as demands. While the inventory is not enough, all the numbers inventory would be taken to deliver tab and the backlog tab will increase with the values from subtraction among demands and inventory. The number of backlog will always increase if the player had a backlog values on the future.

3. Results and Discussion
The database design for the simulation game includes four components, namely parameters, comments, reporting, and help [12]. All variables and decisions are stored in database parameters. This study focuses on designing database parameters as the main component used in simulation applications, other components can be developed afterwards.

The result based on methodology. The first is determining the game's philosophy for later analysis, getting more information about distribution simulation games, and making illustrations of distribution simulation games [8]. Figure 1 shows an illustration of the distribution simulation of a distribution simulation game in the Data Flow Diagram (DFD) model. In the DFD model can be traced every step of the game, data requirements, as well as information from each role. Players are expected to understand the impact of system dynamics represented by random consumer demand, plan inventories to minimize inventory costs and lost sales, determine their demand decisions at the next echelon, thereby acting as managers. This simulation is able to show how decisions from one echelon will affect overall supply chain performance.
Figure 1. Data Flow Diagram For Distribution Simulation Game

As we can see on figure 1, each echelon has a demand called order. The demand will occur to only one echelon above. Therefore, any echelon will not get a same demand from same echelon. Manufacturer is receiving demand from distributor. Distributor is receiving demand from wholesaler. Wholesaler is receiving demand from retailer and retailer will receiving a random number as end customer demand. The number of demands is generated by Monte Carlo simulation especially Linear Congruential Generators (LCG). It is a same with the deliver, one echelon delivering for only one echelon below. Each number of demands will automatically take a number of stocks if the stock covers a number of demands. If is not, the full stock will deliver for covering the demand and the number of backlogs would increase. From this game we will see that the forecasting which we got on
the forecasting course not completely is the same. In course, we have the data demand (for example 12 periods) and we forecast that for next period. However from this game, we learn how to make a forecast by getting demand one by one each period and it is happened in real life. For understanding the relationship of the game, figure 2 shows the Entity Relationship Diagram (ERD) for this game. ERD is an effective communication method between designers and simulation application users as introduced by Paul Fuhs [13].

![ERD Diagram](image)

**Figure 2. ERD Of The Distribution Simulation Game**

Based on ERD, this game has one strong entity and four weak entities and each of them has a role. Each role has similar task and function, which are receiving and delivering order. ERD is base for developing to the next level, the entities and the relationship becomes a key. After ERD, the logical data model created with similar form by ERD. The differences are on detail about each attribute. Data type, length of character and some detail of relationship. Furthermore, the physical data model is created. This model is describing review about entities, attributes, and relationship of logical data model. The physical data model is a base for generating into DBMS (Data Base Management System) such as Microsoft Access, MySQL, or others. Figure 3 shows physical data model of distribution simulation game.
The game starts by determining the title, number of periods, type of game (open or closed) and number of players. The first player determines the role for each team member. The data is stored in the Game Title table. When a new game is created, then the first row in four other tables will be created with the first data being the initial inventory of each role. Each player determines the request to another player in the next echelon, the request data will be stored as an order quantity in the next echelon, while the first player will get the order quantity raised by LCG. If the available stock exceeds or equals the demand quantity, the application will simulate data delivery is the same as request, shipping data will be saved in the previous echelon table as additional stock. If the stock is less than the demand, the application will provide the available stock and record a shortage of stock as a backlog that must be fulfilled period thereafter. The game ends after a specified number of periods [14], the winner of this game is the player who has the minimum cost of inventory costs and lost sales [15].

One of the benefits of a distribution simulation game is to show the bullwhip effect on a supply chain management. Based on Pratiwi's research, the backlog values occurs if there are circumstances where the level of demand is too large and not fixed, not comparable with a small level of sales [5]. Meanwhile, according to Suseno research, the bullwhip effect is indeed common in supply chains but the comparison value must be considered [16]. Bullwhip effect value, which is close to the value of one (1), will be better.

This is when compared with this study; the value used on the graph is the result of the sum of stock values with backlog values. It is a response from the stock that responds to the demand value of each period. An analysis of the bullwhip effect can be obtained by comparing the movement of stock and backlog values in each data table in each echelon. Performance analysis among players' teams can be generated from data grouped according to GameID, while analyzes to compare communication effects can be generated based on data groups for open type games against closed type games.

4. Conclusion
This database design produces a different design from the relational database in general. All data in this game can be saved in one single table. However, because this game is planned to be implemented in Microsoft Access while the game is played by several people simultaneously (multiplayer) then it should be divided into several separate tables. Each player accesses a different table to avoid conflicts in the Microsoft Access table. The data stored in the GameTitle table functions as the game's identity and application controller, while the data stored in the table every echelon records every simulation condition, decision and decision consequences. This data can be used for further analysis in

---

**Figure 3. Physical Data Model For The Distribution Simulation Game**

[Diagram of the physical data model]
accordance with Industrial Engineering scholarship such as forecasting, inventory planning, and supply chain management.

**Acknowledgement**

We would like to thank all colleagues at the Department of Industrial Engineering, Universitas Komputer Indonesia, who have supported to do this research.

**References**

[1] Hidayatno A and Moeis A O. 2013. OMG !, pp. 1–90
[2] Moyano I J M, Rahn J and Spencer R. 2014. The Beer Game Fifth Discip. pp. 27–54
[3] Fuhs F P. 1979. The Design of a Database System to Support Business Simulation and Experiential Learning Insights into Exp. Pedagog. 6, pp. 280–7
[4] Hieber R and Hartel I. 2003. Impacts of SCM order strategies evaluated by simulation-based “Beer Game” approach: The model, concept, and initial experiences Prod. Plan. Control, 14, pp. 122–34.
[5] Pratiwi I, Teknik J, Universitas I, Surakarta M, Fitriadi R, Teknik J, Universitas I, Surakarta M, Hapsari R T, Teknik J, Universitas I and Surakarta M. 2007. Analisa Distribusi Produk Dengan Pendekatan Supply Chain Management dan Aplikasi Beer Game, pp. 29–39
[6] Herring R and Swabey L. 2004. Experiential learning in interpreter education Trakia J. Sci. 2, pp. 36–9.
[7] Hoffer J A, Ramesh V and Topi H. 2011. Modern Database Management (Upper Saddle Rive: Prentice Hall)
[8] Dora S K and Dubey P. 2013. Software Development Life Cycle (SDLC) Analytical Comparison and Survey on Traditional and Agile Methodology Natl. Mon. Ref. J. Res. Sci. Technol. 2, pp. 22–30.
[9] Santosa A and Anugrah S A. 2018. Design of business simulation game database for managerial learning IOP Conference Series: Materials Science and Engineering
[10] RAND CORPORATION. 1960. The Game Monopologs
[11] Valacich J S and George J F. 2017. Modern Systems Analysis and Design ed S Wall (Boston: Pearson Education, Inc)
[12] Hall J J S B. 2004. COMPUTER SIMULATION: A DESIGN ARCHITECTONIC Dev. Bus. Simul. Exp. Learn. 31, pp. 166–75
[13] Fuhs F P. 1988. Event-Extended Entity-Relationship Diagrams For Understanding Simulation Model Structure And Function Dev. Bus. Simul. Exp. Exerc. 15, pp. 41–5
[14] Edali M and Yasarcan H. 2016. Results of a beer game experiment: Should a manager always behave according to the book? Complexity, 21, pp. 190–9
[15] Rafique M Z, Saadat W, Ur A and Mir R. 2014. The Improvements in Supply Chain Management through Beer Game, 4, pp. 1–5
[16] Suseno L and Ikatrinasari Z F. 2016. Analisis Bullwhip Effect Terhadap Penerapan Distribution Resource Planning Di Pt. Mnj J@Ti Undip J. Tek. Ind., 10. pp. 141–8