

Comparison Analysis of Native Database Design with Object Oriented Design

Muhamad Fernandy^{1*}, Khevien Rizkhi Darmawan², Daniel Yeri Kristiyanto³

^{1*,2,3}Department of Information System, Faculty of Informatics, Institut Teknologi Telkom Purwokerto

^{1*}20103134@ittelkom-pwt.ac.id, ²19103096@ittelkom-pwt.ac.id, ³daniel@ittelkom-pwt.ac.id

Abstract

Database design requires a structured database design because the database contains data or information. The design method of the database design determines the structure of the designed design. Database design has two methods, either native or object-oriented method. Native database design has two stages, it is Data Flow Diagram and Entity Relationship Diagram, whereas if it is object-oriented design using use case diagrams. It is accompanied by class diagrams. Native designs tend to be more unstructured than object-oriented design. Native design focuses more on entity flow while object-oriented design focuses on database design entities. Another advantage of using object-oriented design is the ease of explaining the database design to the client because of the simple design so that it can be easily understood. The method used in this research is prototype and relational algebra. The prototyping method is a technique to collect certain information about the user's information needs appropriately. This research focuses on comparing the native and object-oriented design.

Keywords: Database, Database Design, Native Database Design, Object-Oriented, Object-Oriented Database Design

© 2023 Journal of DINDA

1. Introduction

Database design requires a structured database design, because inside the database contains data or information. The process of choosing a database design analysis method is important, because in designing a database, a structured design is needed. There are two methods that can be used in designing database design, namely native and object-oriented. In designing a design using native, first create a Data flow diagram to explain the direction of one entity to another. "The data flow diagram has three levels, namely a context diagram that depicts 1 process that represents a system as a whole and the DFD is labeled zero" [1].

Furthermore, the Data flow diagram level 1 where the next stage of the context diagram explains all the processes in the context diagram in detail so that it is completer and more detailed. The last stage there is a level two data flow diagram which is a detailed data flow diagram of the data flow in the level one data flow diagram which is in charge of outlining what are the process processes that occur at DFD level one. After creating the Data flow diagram then proceed to create an Entity relational diagram to conceptually annotate the graphics in data modeling. Object-oriented design has a use case diagram and use class for the description of the

designer, in making a use case diagram there are several symbols as a representation of the entity, here is the name of the symbol name in the use case diagram, first the actor to represent the role of people, secondly the use case for interaction between actors and the system and for abstraction. Third, association is an abstraction of the relationship of actors and use cases. Included to show that the use case is fully functional. Fourth, Generalizations are used to expose the role of each actor to be able to play a role in the use case, consisting of demonstrating that the use case is entirely the functionality of the other use cases, and the closing exception indicating that the use case is additional. functional from another example of use, if the situation has been met, to model and declare the units of function of the provider provided by using the units in object-oriented design, use training can be used. "This Class diagram shows the structure and description of group packages and objects and correlations with each other similar to containment, inheritance, association and others"[2].

"A database is a row of archives that are interrelated and interact with each other, the relationship if shown with the key of each existing file" [3]. Furthermore, the process of creating a database has several stages that can be done. Design is a stage that serves to describe

relationships and entities in the database. Database design is the process of generating detailed model data from a database. The method used determines the model description of the existing database so that it needs to be considered in the selection of design.

This study aims to compare the design of native database designs with object-oriented. It is hoped that this research can help people to design database designs quickly and efficiently. So, the main purpose in this research is to find the best method.

2. Research Methods

The method used is a prototyping method. "The prototyping method is a technique to collect certain information about the user's information needs appropriately" [4]. The following is an overview of the prototyping method:

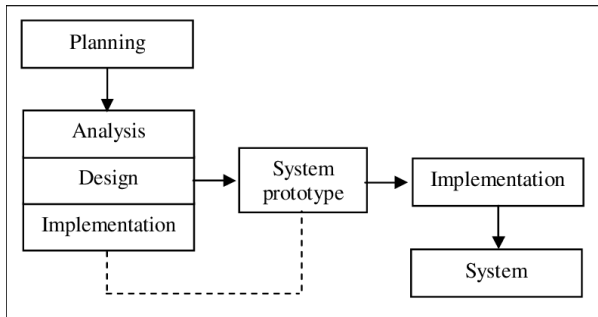


Figure 1. Prototype Method

The Relational Algebraic Operations used in this study are annotated (σ) as selection. The selection operation is an operation that selects a tuple that meets a predicate. Here are the comparison operators that can be used ($>$, $<$, \geq , \leq , \neq , $=$, $\#$). "Multiple predicates can be combined into diverse predicates using the AND (\wedge), OR (\vee), and NOT (\neg) links"[5].

The select process in the student table with the aim of displaying a Nama_siswa named alex fabri can use relational algebra commands with statements as follows:

σ book = 'Sun'(book)

Information:

σ = symbol select

title = 'Sun' = selection condition

(book) = Relation name

The select process on the maple table with the aim of displaying the physics Mapel can use relational algebra commands with statements as follows:

σ name='Muhamad Fernandy'(member)

Information:

σ = Symbol select

name='Muhamad Fernandy' = selection condition

(member) = Relation name

3. Results and Discussion

3.1. Database

"We can know that database is a repository of data. But in theory, a database is a complex set of data or information, a set of data is organized into groups with similar data types called entities), where data can be connected to each other, making it easy to access." [3]. In making a comparison between these two methods, I will use a simple database example from the library. This database will be material to compare design analysis methods that are widely known among IT people, namely structured or native design analysis with object-oriented design analysis. Here is a table from the library database:

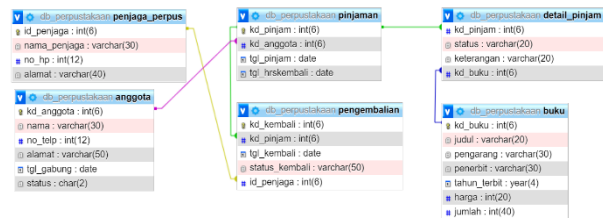


Figure 2. Design of the library database

3.2. Data Flow Diagram

There are 3 levels in the Data flow diagram or DFD. The first level is the Data flow diagram stage 0 or what we usually know as the context diagram. Second, the data flow chart is stage one and the last data flow chart is stage 2 or. "DFD level zero is the top level in the data circulation diagram, which shows the system as a whole." [6]. The following is the Data flow diagram of stage 0 or what we usually know as context diagram 0 from the database library:



Figure 3. Library Database Context Diagram

In the picture above, you can see a database context diagram that is quite simple because there is no task parsing. Data flows from the student's external to the main process of the begin library as well as vice versa. And there is also an external so-called library guard, just like external students, there is an exchange of data

between the external library guard and the library process.

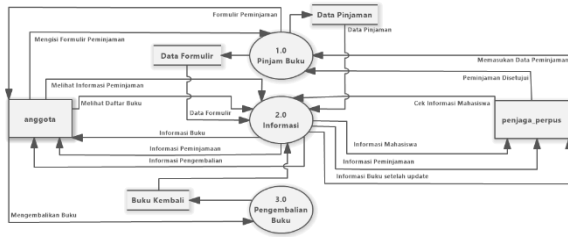


Figure 4. DFD level 1 database library

In the picture above, there is a level 1 data flow diagram that parses the process that was previously carried out together in the context diagram process, now divided again according to the class. "In this process is a context diagram whose data circulation is storage data" [7]. There are 3 processes, namely, borrowing books, information and book returns.

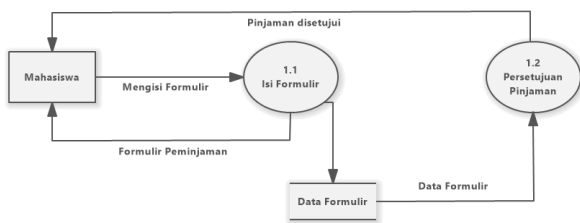


Figure 5. DFD level 2, details of the library database book borrowing process.

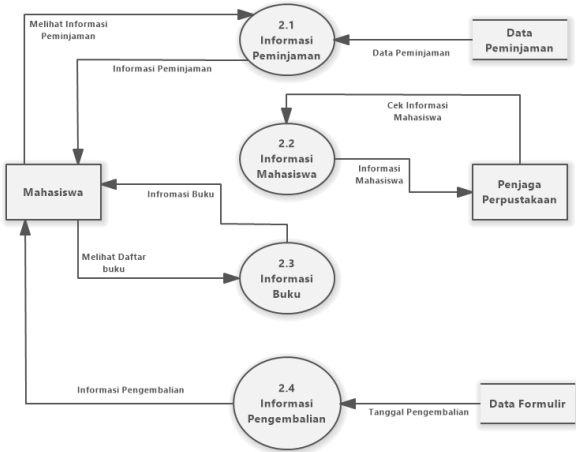


Figure 6. Details of DFD stage 1

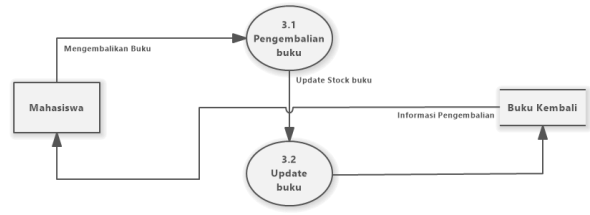


Figure 7. Details of DFD stage 2

3.3. Entity Relational Diagram

Indicates the entities involved in the system and the relationships between these entities. Here is the Entity relational diagram of the Database Library:

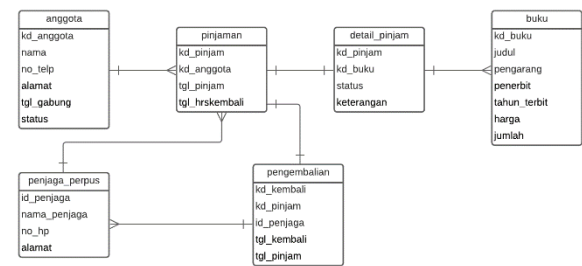


Figure 8. ERD database library

Above is the Entity relational diagram of the data flow that has been created. There are 6 tables and relationships that connect them.

3.4. Undefined Modelling Language

"Unified Modeling Language is one of many visual modeling used in the design and creation of object-oriented software" [8]. The main flow used in the manufacture of UML is the use case diagram. "A diagram describing actors in the use case and their relationships becomes a sequence of actions that conveys measurable value to actors" [9]. The following is a use case diagram of the library database design:

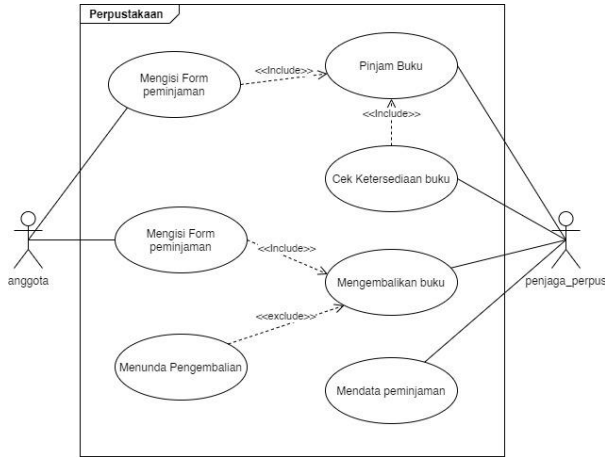


Figure 9. Library Use Care

The above is a use case diagram of the library database. There are 7 use cases, 4 use cases from the library guard interface side and 3 from the member interface side. "Class diagrams are illustrative illustrations of correlations between classes or tables used in the system" [10]. The Class diagram illustrates the structure and description of each class. There are 3 main areas in the Class Diagram, namely: (1) Name (2). Attribute (3). Method. Here is the Class Diagram from the library database:

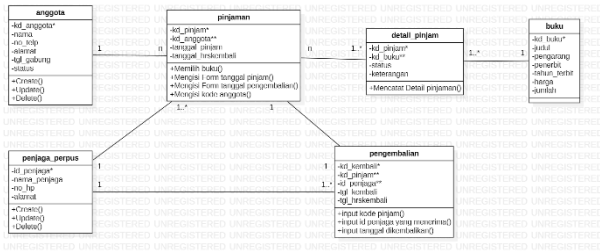


Figure 10. Library database diagram class

The above is a use class of library databases that have 6 tables like those in the database.

3.5 Comparison

Table 1. Comparison of Table

Native	Object Oriented
Use the Data flow diagram to describe in detail each database flow.	Compared to Object oriented which uses a use case, because it focuses on what objectives the subject can do.

Use the Entity relational diagram to illustrate the flow of database entities.	While in object Oriented this uses a use class to describe the attributes of the database.
It doesn't explain what methods each entity uses.	In the use class, explain what methods are used.
There is no visibility in each attribute used.	There is visibility in the use class so that database developers can know the access rights of each database attribute.

From the second use of database design in library databases, with native-based design using DFD to describe the flow and ERD for attribute details. In object-based database design, use cases can be used instead of DFD to describe the flow, but focus more on objects and for ERD replacements, can use classes that not only describe attributes but also methods to be used on those entities.

4. Conclusion

Based on the results of the implementation in the library database, the analysis of native design is compared with the analysis of object-oriented design. It can be concluded that design analysis by orienting objects is superior to structured or native design analysis. Object Oriented Programming is superior because it tends to be more able to minimize the design time and can be said to be easier to understand by ordinary people, so that in presenting to the client, there is no need to make design designs that are easy to understand because the design designs that have been made using object-oriented methods are easy to understand by non-IT people. Based on these conclusions, in analyzing and designing database design, it is recommended to use object-oriented methods

References

- [1] Y. Suro, "Jurnal Ilmiah Universitas Batanghari Jambi Vol.14 No.4 Tahun 2014 DATA FLOW DIAGRAM (DFD) PADA APOTEK CANDRA KOTA JAMBI Yunan Suro 1," *J. Ilm. Univ. Batanghari Jambi*, vol. 14, no. 4, pp. 56–64, 2014.
- [2] Agus Suryana, "Unified modeling language," *Perform. Comput. Rev.*, vol. 14, no. 13, 1996, doi: 10.4018/jdm.2001010103.
- [3] R. S. dan J. Febio, "MEMBANGUN APLIKASI E-LIBRARY MENGGUNAKAN HTML, PHP SCRIPT, DAN MYSQL DATABASE Rini Sovia dan Jimmy Febio," *Processor*, vol. 6, no. 2, pp. 38–54, 2011.

- [4] RANI SUSANTO, “PERBANDINGAN MODEL WATERFALL DAN PROTOTYPING UNTUK PENGEMBANGAN SISTEM INFORMASI,” *Ther. Ggw.*, vol. 46, no. 5, pp. 756-757 passim, 2016, [Online]. Available: <https://repository.unikom.ac.id/30459/1/4.miu-14-no-1-rani.pdf>.
- [5] D. Setiyadi, “Aljabar Relational dan Implementasi kedalam Bahasa Query dalam Perancangan *Database* Relational,” *J. Kaji. Ilm.*, vol. 20, no. 2, pp. 213–224, 2020.
- [6] K. Hapsari and Y. Priyadi, “Perancangan Model *Data flow diagram* Untuk Mengukur Kualitas Website Menggunakan Webqual 4.0,” *J. Sist. Inf. Bisnis*, vol. 7, no. 1, p. 66, 2017, doi: 10.21456/vol7iss1pp66-72.
- [7] A. R. Simatupang and S. Nafisah, “Analisis Proses Pada Senayan Library Information Management System (SLIMS) Cendana Berbasis *Data flow diagram* (DFD) Di Perpustakaan Universitas Kristen Duta Wicana Yogyakarta,” *JUPI (Jurnal Ilmu Perpust. dan Inf.*, vol. 5, no. 1, pp. 1–15, 2020.
- [8] M Teguh Prihandoyo, “Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web,” *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 126–129, 2018.
- [9] Havaluddin, “Memahami Penggunaan UML (Unified Modelling Language),” *Memahami Pengguna. UML (Unified Model. Lang.*, vol. 6, no. 1, pp. 1–15, 2011, [Online]. Available: <https://informatikamulawarman.files.wordpress.com/2011/10/01-jurnal-informatika-mulawarman-feb-2011.pdf>.
- [10] Suendri, “Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan *Database* Oracle (Studi Kasus: UIN Sumatera Utara Medan),” *J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, pp. 1–9, 2018, [Online]. Available: <http://jurnal.uinsu.ac.id/index.php/algorithm/article/download/3148/1871>.