

RELATIONAL DATABASE STRUCTURE AND OPERATIONS ENGINEERING USING CLASS DIAGRAM AND ACTIVITY DIAGRAM

Lukman Hakim

School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
23515045@std.stei.itb.ac.id

M. Sukrisno Mardiyanto

School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
sukrisno@informatika.org

Abstract— Software system plays an important role especially for processing large amounts of data. Currently, the operation of a data is mostly done on the application layer, so the DBMS only serves to store data processing results. Data processing on the DBMS layer will make better performance, as it is closer to the physical data. Application of constraints as well as data access rights at the DBMS level will apply consistently to the system as a whole so as to maintain data integrity. In this research, UML-based approach is proposed to map the structure and operation in a domain model into the structure and operation of relational database. A software will be built using a driven architecture model (MDA) to automate the transformation from UML to SQL script. Class diagrams that describe the domain structure of the model will be mapped into the table structure on the relational database, while the Activity diagram that describes the behavioral aspects of a domain model will be mapped into procedure and function. It is expected that this research can simplify the process of database schema development, increase data processing speed and maintain data integrity.

Keywords; *UML Diagram, Model Driven Architecture, Relational database, Code Generator*

INTRODUCTION

Software is built to provide an interface that allows users to perform operations into the database system. So there must be compatibility between software design and database design. The current software is mostly built using an object-oriented programming language modeled using UML diagrams, but on the other hand the DBMS used is a relational model. The operation of a data is mostly done on the application layer so that the DBMS only serves to store the data processing results. According to Andersuu and Mircea [4], data processing on the DBMS layer will make better performance, as it is closer to the physical data.

This research proposed will expand usage of DBMS, not only to store the results of data processing, but also to handle the operation of the data, as well as to control the operation of a data. It can be done by building a software to transform UML diagrams into scripts that can run on the DBMS.

This research uses Model Driven Architecture (MDA) which consists of 3 layers: Computational Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM) with input of XMI file that representing UML diagram. UML modelling is done using papyrus extension on Eclipse IDE. The modeling results are then stored

as XMI file. The pre-processing module parses XMI file based on certain tags according to the object want to build

CIM layer will be used to define entity entities and processes that exist in the database schema, at this layer, detail of how a process will be run has not been defined. The diagram used in the CIM layer is the class diagram.

The PIM layer will be used to build an activity diagram object that contains a detailed representation of an operation in a class diagram. PSM layer will be to define rules for transforming the model that has been generated on the PIM layer. In this research, the model generated from the PIM layer will be transformed into a MySQL script.

I. RELATED WORK

Various works in the field of UML to relational data modeling have been conducted during the last few years using various techniques. However, almost all of the previous research focus on static schema, none of them discuss about operations at database layer.

Pavkovic and Brdjanin [1] discuss a method for forward engineering of SQLite database. This research proposed an approach based on the standart UML Notation for conceptual modelling and for representation of the target relational database schema.

Yang [2] proposed a database modeling method on Enterprise Management Information System (MIS) using UML diagrams. This research discusses how to model the database structure, especially how to manage the complexity of relationships between tables. This research does not discuss how to map operations on a database.

Chih-Min Lo and Hsiu-Yen Hung [3] proposed adding additional notation to UML diagrams to model the structure and retrieval of data in the database. The addition of new notation to the UML diagram causes this approach not necessarily acceptable to all parties. This research uses sequence diagrams to model operations on the database. Though sequence diagram is more appropriate used to model interaction between object, not detail of an operation

II. UML DIAGRAM METAMODEL

The Mapping of UML Diagram required a metamodel that represents the UML diagram structure. Figure 1 shows the UMLComponent metamodel. This model can be derived for class diagrams and activity diagrams.

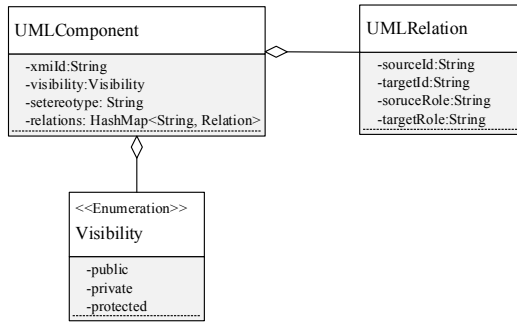


Figure 1. UML metamodel

Figure 2 shows the structure of UMLClassDiagram that used to represent a class diagram, this class is a derivative of the UMLComponent class

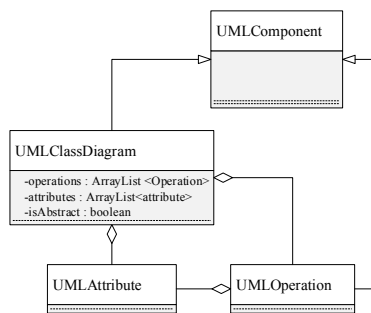


Figure 2. UML metamodel

Activity diagrams are used to model the details of an operation in a class diagram. Notation in the activity diagram is divided into two parts: nodes and edges. The node consists of two actions node and control node. Action node is used to execute the statement contained in its body, while the control node is used to set the order of execution node. Control node consists of several types of initial node, decision node, merge node and final node. Figure 3 shows a metamodel that describes the structure of the Action node and Control Node, both of them are derivative of UMLComponent.

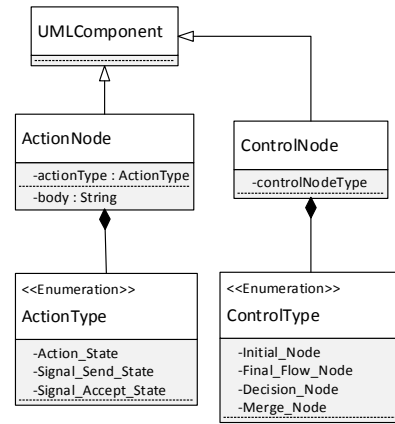


Figure 3 Activity metamodel

III. MAPPING OF OBJECT ORIENTED CONCEPT TO RELATIONAL MODEL

A. Inheritance Mapping

In this research, only concrete classes will be mapped into tables, therefore there is no association between tables. The advantages of this method are efficient in storage.

In the simple class diagram, the attribute will be mapped into columns in the database table. However, in the class diagram that has inheritance, the parent class attribute will be added to the mapping result table of the child class.

As an example of Figure 4 the Karyawan class is an instance of the class of Orang, therefore the employee class mapping table will contain columns derived from the class attribute of the person.

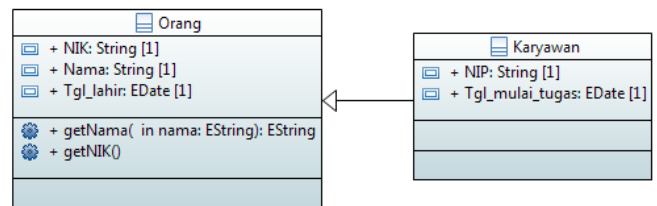
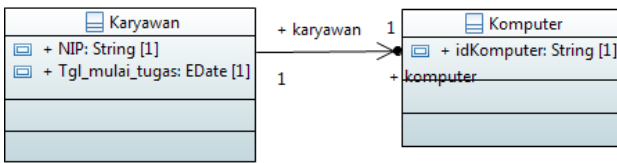


Figure 4. Inheritance example

B. Association and Compositioan Mapping

Associations will be mapped to relationships between tables, eg class A associated with class B, then the resulting table is table A which has columns referring to one of the columns in class B. Figure 5 shows an example of association relation.



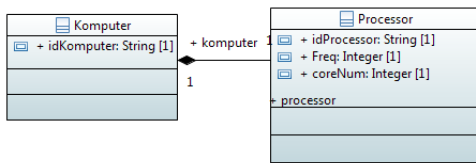
```

Create tabel Karyawan
(
NIP String, Tgl_mulai_tugas Date
PRIMARY KEY ( NIP),
CONSTRAINT _x70MAAhjEeiyAN4OFXww6w
FOREIGN KEY (
idKomputer) REFERENCES Komputer (idKomputer) )

```

Figure 5. Association mapping result

Compositions will be also mapped to table relationships. there is an additional constraint ON DELETE CASCADE ON UPDATE CASCADE to maintain reference integrity.



```

Create tabel Karyawan
(
NIP String, Tgl_mulai_tugas Date, NIK String, Nama
String, Tgl_lahir Date, idKomputer String
PRIMARY KEY ( NIP),
CONSTRAINT _x70MAAhjEeiyAN4OFXww6w
FOREIGN KEY (
idKomputer) REFERENCES Komputer )ON DELETE
CASCADE ON UPDATE CASCADE

```

Figure 6. Association mapping result

IV. MAPPING OF OBJECT ORIENTED CONCEPT TO RELATIONAL MODEL

A. Decision Block Mapping

Tag <node> will be parsed to find control nodes and action nodes in the XMI file. The data stored in the <node> tag is used to build control node object. Control nodes and Action nodes have been built are not equipped with execution sequence. In XMI files, the execution sequence is stored in the <edge> tag. The result of parsing the <edge> tag will be represented in Node List. A Relation object will be formed for each node that be found. The Relation will control execution sequence of action node.

Activity diagram mapping is started by convert decision node to action node. A deep-first search will be performed to find a pair of decision node and merge node that indicate a decision node block. That block will be convert to action node with IF-ELSE code written in its body as shown in figure 8 .

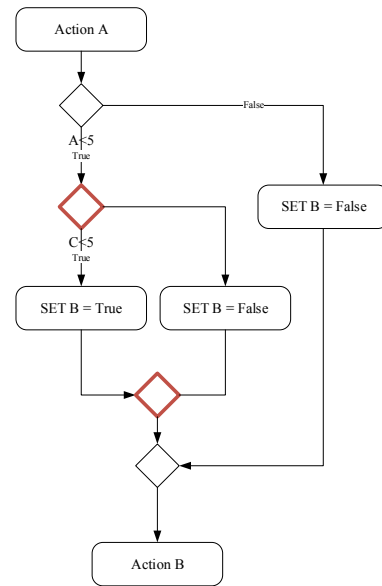


Figure 7. Nested decision block

Mapping process will be continued to remaining decision-merge node pair. Mapping process will be ended when all of decision-merge node pair have been convert to action node as shown in figure

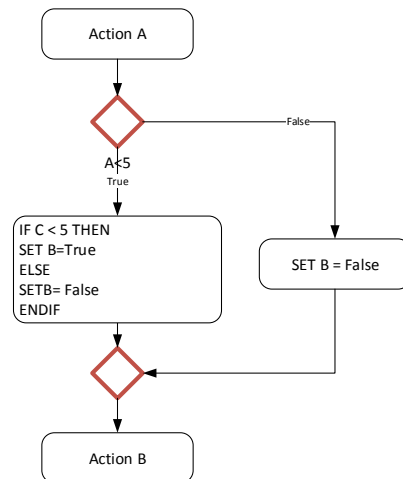


Figure 7. Decision block mapping result

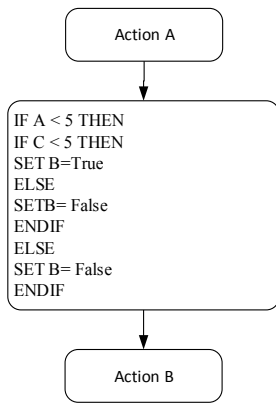


Figure 9. Final result of decision block mapping

B. Looping Block Mapping

Activity diagram that used in this research are not supported loopin notation. A loop can be formed when the process flow from a decision node block returns to the beginning of the decision node again after running some processes. As shown in figure 10 , the process flow on the red arrow returns to the beginning of the decision node after running some processes, therefore the activity block is detected as looping.

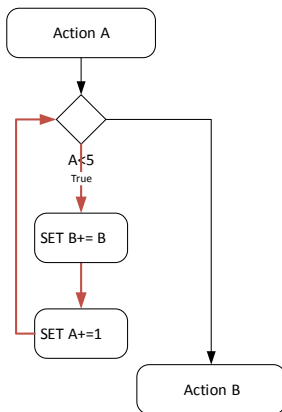


Figure 10. Looping block

Figure 11 show an example of nested loop. The deepest looping block will be converted to be an action node with looping seouce code within its body. The process then continued to next deepest looping node. The process is ended when all of looping block have been converted to be action node.

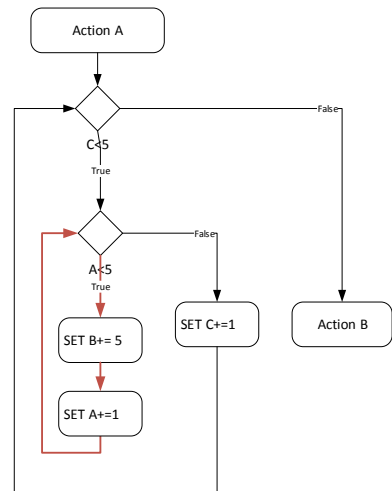


Figure 11. Nested decision block

Figure 12 shows the final result of nested loop processing. An action node will be formed in place of the looping block. The program code representing the previously processed decision node will be inserted in the body of the action node.

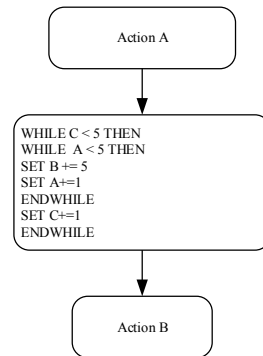


Figure 12. Looping block processing final result

V. EVALUATION

Tests is conducted on two applications namely library information system and academic information system developed by JIBAS (Jaringan Informasi Bersama Sekolah). The following table is the specification of each information system used.

Table I. Testing environment spesifications

Aspect under test	System under test	
	Library System	Academic Information System
Aggregation	Yes	Yes
Association	Yes	Yes

Aspect under test	System under test	
	Library System	Academic Information System
Composition	Yes	Yes
Inheritance	Yes	Yes
Number of classes	13	47
Number of operation	5	10
Mapping accuracy	98%	90%

Performance testing was also performed. This test to see comparison of processing speed between stored procedure with data processing in programming language such as Java and PHP. Testing was performed by calculating the average score of 9 lessons from 1980 students. When testing, MySQL server, Webserver or JDK Java are in the same machine. From the results of the test it appears that stored procedures faster in data processing, even when located on the same machine with Java and PHP. However the stored procedure speed will be much decreased when performing complex calculations.

Table II. Performance testing result

Program	Number of data	Execution time
Stored Procedure	1980	0.04 detik
Java	1980	0.7 detik

Program	Number of data	Execution time
PHP	1980	0.08 s

VI. CONCLUSION AND FUTURE WORK

By this research, we know that class diagrams and activity diagrams can be used to build structures and operations on the database automatically.

Performance testing shows that data processing of query results on the DBMS layer are faster than data processing using programs outside the DBMS, even while on the same machine, but complex arithmetic operations on the DBMS is significantly slower than processing using programs outside the DBMS.

Future refinements of work involves expanding roles of OCL, and trigger to maintain data security and integrity, In current version, OCL is used just for defining filed specification.

REFERENCES

- [1] Pavkovic, Z. S. dan Brdjanin, D. (2016) : A UML-based approach to forward engineering of SQLite database : Zooming Innovation in Consumer Electronics International Conference, 80 – 83
- [2] Yang, J (2010): Database Modeling of Enterprise Management Information System based on UML : Computational Intelligence and Software Engineering, 1 - 4
- [3] Lo, C. M., dan Hung, H. Y. (2014) : Towards a UML Profile to Relational Database Modeling : Applied Mathematics & Information Sciences International Journal, 8, 733 - 743
- [4] Anderscu, A. I. dan Mircea, M. (2012) : Perspectives on the Role of Business Rules in Database Design : Database Systems Journal, 3, 59 – 67