

Eloquent Object Relational Mapping Models for Biodiversity Information System

Edy Budiman¹, Muh Jamil², Ummul Hairah³, Hario Jati⁴, Rosmasari⁵
dept. of computer science and information technology
Universitas Mulawarman
Samarinda, Indonesia
edy.budiman@fkti.unmul.ac.id

Abstract—Enterprise-scale applications are developed with an object-oriented approach, meanwhile, the widely used database management system now is a relational database, the mismatch between relational database and application development of the object-oriented concept is a problem between object-oriented environment with environment Relational Database Management System (mismatch impedances). This paper discusses the implementation of Eloquent Object Relational Mapping (ORM) for developing of Biodiversity Information System (BIS) and Model View Controller (MVC), which focuses on mapping the relations medicinal, wood, and tree object in the form of associations; One to One relationship, One to Many, Many to Many, Polymorphic Relations, Many Through Many, Many to Many Polymorphic Relations. The results of the implementation explain that the six concepts of association owned by Eloquent ORM can facilitate complex data relationships such as BIS by simplifying complex code writing to relate between tables by relating the class model so that it can be a solution to deal with complex data cases such as BIS. Eloquent ORM can mediate between the application layer and the database environment by transforming data from the relational database environment into object form and vice versa.

Keywords—*eloquent; object-relation-mapping; biodivesity; model-view-controller;*

I. INTRODUCTION

The technological revolution of web application development in the framework concept is increasing rapidly [1], [2], many vendors and developers have created framework models with various features and advantages possessed [3], such as Yii [4], CodeIgniter [5], Symfony [6], Laravel [7], and others. Other studies have also previously conducted a feature comparison review that is shared by various frameworks such as [8-9], [3], and one of the technologies of the framework is the ability to map database objects, with key features that can bridge the data source environment with the application environment [10].

Biodiversity Information System is an enterprise information system that represents taxonomic data, forms an ontology model that can present, map data as information about data descriptions and relationships between taxon according to taxonomic level based on data stored in the database [11].

Development of Biodiversity Information System (BIS) which has been developed by [12-13] in its development using object oriented approach. Meanwhile, the database management system that many taught is the concept of relational database, while according to [14] enterprise-scale application development will be more easily developed if using the concept of object-oriented because it can reduce the writing of long code and also support the reuse of source code from other parties if time is needed.

Thus, there is a mismatch between relational database concepts and application development using object-oriented concepts. And that according to [15] is an issue between an object-oriented environment and the RDBMS (Relational Database Management System) environment often referred to as impedance mismatch [16]. Such discrepancies include, among others, aspects of granularity, subtypes, identity, association and data navigation [16-18].

To overcome this, it takes a method of mapping entities (tables) that exist in the database into an object, Relational Mapping Object (ORM) is one method of programming that can translate object-oriented concepts (classes, methods and attributes) into the concept of relational database (Tables, columns and values) [10].

There are many types of ORMs, and one of them is the Eloquent ORM that has the Laravel Framework [19], [7]. Eloquent ORM in [7] contains six association concepts that can be used in order to overcome the difference of object-oriented paradigm with the relational paradigm of databases in terms of data associations.

In this paper will discuss the implementation of the six data association concepts that exist in Eloquent ORM so it can be one solution in handling data cases that have complex complexity of relationships such as Biodiversity Information System, using the Framework Laravel as a framework that has an MVC design pattern that will provide Its own convenience in terms of system development that embraces the object-oriented paradigm to mapping data from Relational Database Management system (RDBMS) to the application layer.

II. RELATED WORKS

A. ORM (Object Relational Mapping)

Research related to ORM has been widely practiced in a variety of contexts and issues, including [20] which addresses the approach of natural interface language to relational databases using ORM, Alexandre Torres, et al [21] which addresses the work patterns of some ORMs to address developmental discrepancies Application with object-oriented paradigm with relational paradigm, and Scott W. Ambler [22] which discusses the mapping objects to relational databases: O/R mapping in detail. The whole of this research leads to one problem that is the occurrence of object-oriented paradigm mismatch with relational which is often called mismatch impedances.

B. Laravel Framework

Laravel is the first PHP framework released on June, 9, 2011 with the initial version of version 1, the update continues to be done from year to year with the addition of various features such as the addition of DMBS support and the addition of migration feature to database schema migration to support creation of relations between classes Modeled on the ORM [23] and until this study was made there has been a revised version of Laravel Version 5.4.

Similar to other PHP framework concepts, Laravel is built on the basis of MVC (Model-View-Controller), and also has ORM technology called Eloquent, Eloquent ORM. It is a feature that implements active record patterns that can simplify the Create, Update, Retrieve, and Delete processes of the database records. Eloquent ORM has 6 relationship methods as described in [7], i.e.; *One to One*, *one to Many*, *Many to Many*, *Has Many Through*, *Polymorphic Relation*, and *Many to Many Polymorphic Relation*.

C. Biodiversity Information System (BIS)

Reference [24] explained that diversity is a complex concept and the term is used in many ways, and most generally according to [24] "diversity refers to the total number of species inhabiting an area. and according to US. Congress Office of Technology Assessment [25] "Biological diversity or biodiversity" will be used when we mean the variety and variability among living organisms, and the ecological complexes in which they occur.

BIS an information system that presents biodiversity data that its distribution in a particular area, is an enterprise-scale information system that represents taxonomic data, establishes an ontology model that can present, map data as information about data descriptions and relationships between taxon according to taxonomic level based on data that are stored in the database.

Examples of biodiversity information systems as developed by the University of Informatics Research Department's research team related to BIS for plant data such as trees and timber [12], and medicinal plants [13] from tropical rainforest Borneo. In the discussion of this paper, the data of medicinal, wood, and tree objects that are contents of diversity information systems will be examples of the application of

Eloquent ORM in mapping and relating data using object-based concepts.

III. DEVELOPMENT METHODOLOGY

This study uses an object oriented approach, which focuses more on data mapping and data relation from DBMS to application level using MVC Framework concept Laravel and Eloquent ORM model as its tools.

A. Object Data and Data Source

The implementation of Eloquent ORM will be applied to Biodiversity Information System which has a lot of complexity of relations. Data to be examples of objects are Medicinal Plants, Tree, and wood data, the data structure can be seen in "Fig. 1".

borneo_medical	borneo_tree	borneo_wood
@id : int(10)	@id : int(10) unsigned	@id : int(11)
@leaf_flower : tinytext	@tree : varchar(255)	@wood : varchar(255)
@rod_root : tinytext	@latin : varchar(255)	@botanis : varchar(255)
@fruit_seed : tinytext	@sinonim : varchar(255)	@local : varchar(255)
@chemical : tinytext	@local : varchar(255)	@sinonim : varchar(255)
@informationresearch : longtext	@picture : text	@habitus : tinytext
@khasiat : longtext	@habitat_ekologi : varchar(255)	@picture : varchar(255)
@deskriptio : mediumtext	@endemik : enum('0','1')	#tree_id : int(11)
#wood_id : int(11)	@tinggi_pohon : varchar(255)	
	@warna_batang : varchar(255)	
	@warna_getah : varchar(255)	
	@created_at : timestamp	
	@updated_at : timestamp	

Fig. 1. Data structure of medicinal, tree and wood tables

B. Methodology for Eloquent ORM

The implementation of the Eloquent ORM model will mapping and relate objects to the medicinal, tree and wood tables. Relationship scheme as follows:

1) *One to One*; this type of relation will be implemented on Medicinal and Wood tables where for one Medicinal data will be related to one Wood data, from these two data will be made One to One relation in each model class.

2) *One to Many*; the concept of this relationship will be implemented on the medicinal and wood, where for each Wood data will have many benefits from the Medicinal table. The model class of these two tables will be one to Many.

3) *Many To Many*; the concept of this relation will be implemented on medicinal and wood table data, where for every wood and medicinal data can have many data that relate to each class of relationship model Many to Many.

4) *Has Many Through*; the concept of a relationship that will be made is a Tree model class can call data on the Medicinal table with Wood table intermediaries.

5) *Polymorphic Relation*; the concept of this relationship will be implemented in the Tree, Wood, and Medicinal tables, the concept of the relationship will be made is, Tree and Wood model class can access data on the Medicinal table in one association.

6) *Many to Many Polymorphic Relation*; the concept of this relationship will be implemented in the Tree, Wood, and Medicinal tables, the concept of the relationship will be made is the Tree and Wood model class can access many data at once from the Medicinal table.

C. Architecture MVC Framework Laravel

The process of mapping data from the database environment to the application environment using MVC Laravel Framework concept can be seen in "Fig 2".

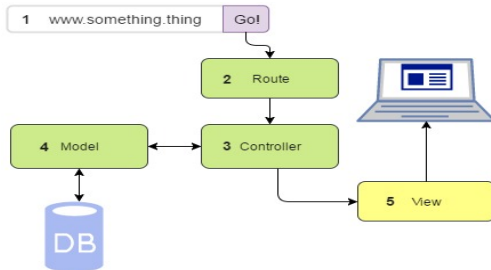


Fig. 2. Laravel MVC Framework Concept

Session 1), the URL input form in the browser, 2) the Route, where the URL that has been input to the form will be matched with the existing URL present on the route, 3) the route will access certain methods on the controller class and the controller will then call the class model according to the model class name that has been defined, 4) the model class will access the data in the biodiversity database and the data that has been taken will be returned on the controller class to be displayed on the view, 5) the view will display the Medicinal, Tree or Wood data provided by the controller and then the result of the relation mapping can be displayed in the browser window.

IV. IMPLEMENTATIONS AND RESULT

A. Mapping Relation Class Model

In "Fig. 3" shows the data relation scheme of Medicinal, Tree and Wood tables, the data is used as the implementation object of Eloquent ORM implementation.

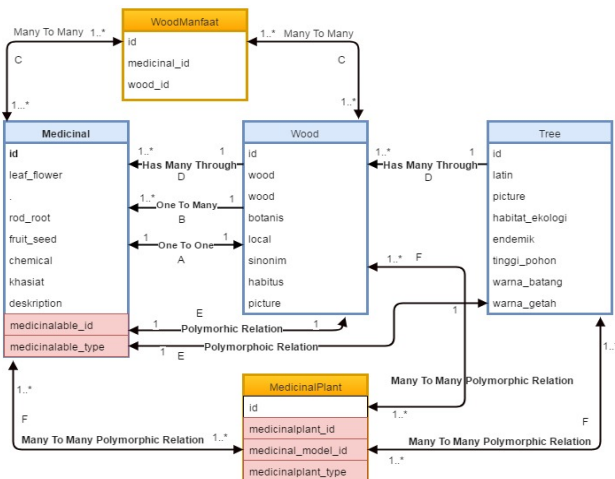


Fig. 3. Relation scheme on Medicinal, Wood and Tree tables

Based on "Fig. 3" the model can be explained in the form of the command of each type of relation for wood, tree, and medicinal data, as presented in TABLE I.

TABLE I. TYPE RELATION AND SYNTAX COMMAND CLASS MODEL

Relation Type	Command Class Model
One To One	<pre>public function wood(){ return \$this-> hasOne('App\Biodiversity\WoodModel','medicinal_id'); } public function medicinal(){ return \$this-> belongsTo('App\Biodiversity\MedicinalModel'); }</pre>
One To Many	<pre>public function medicinal(){ return \$this-> hasMany('App\Biodiversity\MedicinalModel','wo od_id'); } public function wood(){ return \$this-> belongsTo('App\Biodiversity\WoodModel'); }</pre>
Many To Many	<pre>public function wood(){ return \$this ->belongsToMany('App\Biodiversity\MedicinalModel', 'manfaatwood','wood_id','medicinal_id'); }</pre>
Has Many Through	<pre>public function medicinal(){ return \$this-> hasManyThrough('App\Biodiversity\MedicinalModel', 'App\Biodiversity\WoodModel','tree_id','wood_id'); }</pre>
Polymorphic Relation	<pre>public function medicinalable(){ return \$this->morphTo(); } public function medicinals(){ return \$this-> morphMany('App\Biodiversity\MedicinalModel', 'medicinalable'); } public function medicinals(){ return \$this-> morphMany('App\Biodiversity\MedicinalModel', 'medicinalable'); }</pre>
Many To Many Polymorphic Relation	<pre>public function medicinalplant(){ return \$this-> morphToMany('App\Biodiversity\MedicinalModel', 'medicinalplant'); } public function treemedinal(){ return \$this ->morphedByMany('App\Biodiversity\TreeModel', 'medicinalplant'); }</pre>

Writing command code (syntax) in the class model of the six existing types of relationships shows that to perform the process of relation model Eloquent ORM simply by writing the type of relation that will be used, then write the name of the class model to be related and references key used. Code writing for a relation is done once and can be used repeatedly with the same form of data and relations.

Meanwhile, if using the concept of the relational database requires a longer and more specific command of each table name to be correlated, and must specify the attributes to be called from each table and of course must mention the name of

the primary key and foreign key parameters in the creation A relation between tables.

The form of writing a relation with the type of relationship One to One on medicinal data with wood data by using the concept of relational database presented in "Fig. 4.

```
SELECT wood.id, wood.wood, wood.botanis, wood.local, wood.sinonim, wood.habitus, medicinal.leaf_flower, medicinal.rod_root, medicinal.fruit_seed, medicinal.chemical FROM medicinal LEFT JOIN wood on medicinal.wood_id = wood.id WHERE medicinal.id=1;
```

Fig. 4. Query SQL Medicinal and Wood data relation

"Fig. 4. shows the SQL Query to relate and display medicinal data relating to wood data with One to One relation type. It appears that the table name, attribute, and foreign key and primary key names should be written specifically to perform a relation.

The writing of SQL syntax would be different if the relationship concept of these two data is reversed, but still not simplify the writing of its SQL syntax, Writing SQL syntax will become longer and more complicated if the case of data encountered more complex, of course, SQL query will continue to be repeated if the data calling is done on another application view with the same data case and relation.

While compared to the relation method by using Eloquent ORM simply by adding the *hasOne* command to the medicinal model class, followed by writing the name of the model class to be correlated, followed by writing the references key name used. If the concept of the relationship between these two data reversed the writing of the relational, command simply by adding the *belongsTo* command to the wood model class and followed by writing the name of the medicinal model class.

Actually, in the process of calling the data and its relation, the command contained in Eloquent ORM will be generated into a series of SQL commands. However, the existing SQL command has been shortened to be easier to remember so that the concept offered by Eloquent ORM is easier in terms of writing because it does not have to mention the name of tables and attributes in specific but simply write the name of the model class that represents the table, thus, of course, will accelerate the developer in The construction of a system with a case of complex data relations.

B. Mapping in MVC Framework Laravel

This session, the process will be done mapping data from the RDBMS environment to the application level using the concept of MVC Framework Laravel, the process scheme that occurs can be seen in "Fig. 5".

• **Routing**; In this section, the user will input a URL in the browser form, after the user submits process this URL will then be matched with the existing route list in the program's source code, if the URL is found on the route, the destination URL will throw the command to the controller.

• **Controller**; The controller will then receive commands from the route and match the methods they have with the commands given by the route, then the controller will access the Medicinal model class on the "App\Biodiversity\

MedicinalModel" package, after the class model is accessed then the model class will perform a process calling mechanism On the database record.

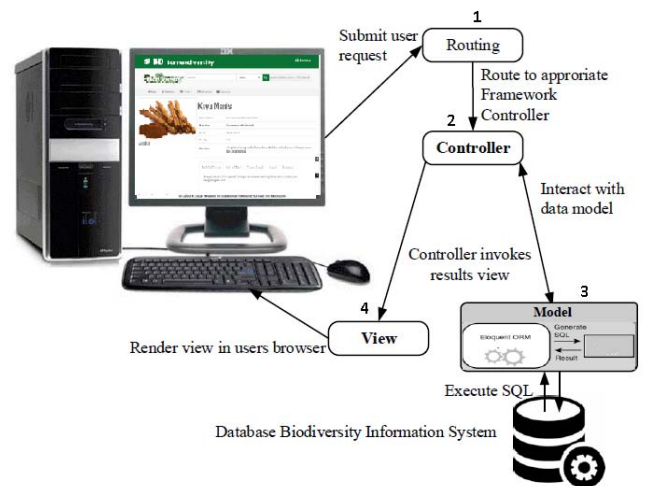


Fig. 5. Mapping of Laravel MVC Framework data

• **Model**: In this section, the medicinal model class will perform the process of relation with the wood model class, with the existing relation command then the wood model class will perform the process of calling data on the wood table on the database in accordance with the relation of the medicinal data. If the relation command has been completed in execution and has a result, then the result in the form of the data record of the database will be given back to the controller to be displayed on the view.

• **View**; In the inner layer the results of command processing and data will be displayed, wood data related to medicinal data and medicinal data itself then displayed on the view with the shape and style in such a way that can be seen by the user. Display User interface mapping data results can be seen in "Fig. 6".

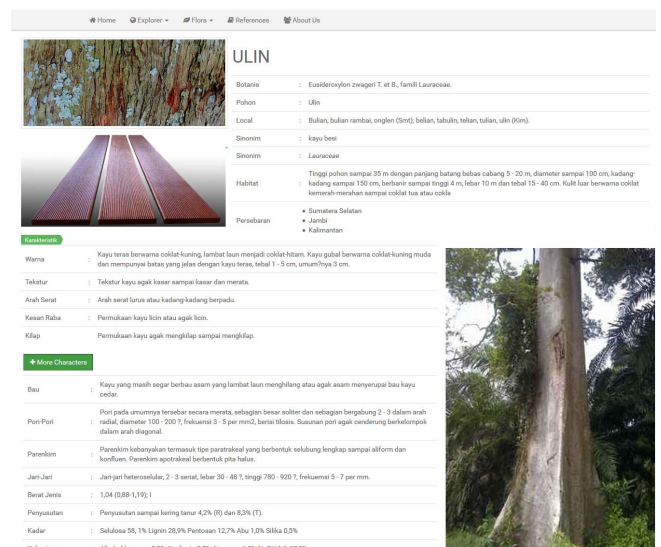


Fig. 6. Screenshot of results output the mapping data polymorphic relation

Based on this explanation, the implementation of the MVC concept in biodiversity information systems offers advantages in terms of easy to use application layer structure management as it separates the logic part (Controller), the part that performs the process of calling data to the database (Model) and the part that becomes the interface with User (View). So that if there is a mistake or damage to the program code then the errors in the structure of the program can be easily fixed.

Another thing that is the advantage of this MVC system is reusable program code in the same data case, so it can accelerate the development of the system by avoiding the writing of repetitive program code with the same function so that it can facilitate the developers in the development of applications that Enterprise.

V. DISCUSSION AND CONCLUSIONS

One approach model to solving complex data association problems such as the Biodiversity Information System using Eloquent ORM Laravel as a tool in mapping and relating data. Eloquent ORM can be a solution to overcome Mismatch Impedance in terms of data associations by implementing an object-oriented association system, so that code writing becomes more concise and reusable.

Interesting in this study is the use of the six types of associations in Eloquent ORM Laravel which has the advantage in handling association problems in complex data cases with the writing of the program code is more concise and simple, but there are still things that need to be studied in depth about the concept of optimization ORM Framework is expected to be able to handle the problem between object-oriented concept and relational concept which is often called Mismatch Impedance which consists of granularity, subtypes, identity, association and data navigation.

ACKNOWLEDGMENT

Thanks to all academic civitas dept. of Information and Communication Technology, thank also to the dean of the Faculty of Computer Science and Information Technology Mulawarman University or support and assistance until this research can be completed properly.

REFERENCES

- [1] A. Brem, and E. Viardot, "Revolution of Innovation Management: Internationalization and Business Models," In *Revolution of Innovation Management*, Palgrave Macmillan UK, 2017, pp. 1-13.
- [2] J. E. Stiglitz, "The Fundamental Flaws in the Euro Zone Framework," In *The Euro and the Crisis*, Springer International Publishing, 2017, pp. 11-16.
- [3] N. Prokofyeva, and V. Boltunova, "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems," *Procedia Computer Science*, vol. 104, 2017, pp.51-56.
- [4] Yiiframework,"Yiiframework official website" www.yiiframework.com. Available at: www.yiiframework.com
- [5] CodeIgniter, "CodeIgniter's official website" codeigniter.com. available at: http://www.codeigniter.com
- [6] Symfony. "Symfony's official website" symfony.com. Available at: http://symfony.com
- [7] Laravel, "Laravel: Official Website documentation" laravel.com. Available at: http://laravel.com.
- [8] E.V. Sandin, N.M. Yassin, R. Mohamad. "Comparative Evaluation of Automated Unit Testing Tool for PHP Framework," *International Journal of Software Engineering and Technology*, vol. 2. No. 2, February 2017.
- [9] J. Samra, *Comparing Performance of Plain PHP and Four of Its Popular Frameworks*, Thesis Project, Linnaeus University, Sweden, 2015.
- [10] A. Decan, M. Goeminne, T. Mens, "On the Interaction of Relational Database Access Technologies in Open Source Java Projects," arXiv preprint arXiv:1701.00416, January 2017.
- [11] J. L.C. Santos, *A biodiversity information system in an open data/metadatabase architecture*, University of Twente, 2003.
- [12] N. Dengen, E. Budiman, M. Ugiarto, J. A. Widians, W. Wati, and U. Hairah, "Biodiversity Information System: Tropical Rainforest Borneo Based on Traditional Knowledge Ethnic of Dayak," 2017 Progress in Electronics Engineering, Computer Engineering and Information Technology (PIECT), Bandung, Indonesia, May 2017.
- [13] E. Budiman, A. Tejawati, U. Hairah, and Haeruddin, "Biodiversity Information System of Medicinal Plants from Tropical Rainforest Borneo Based on Traditional Knowledge Ethnic of Dayak," 2017 ICOCOE Series 4th International Conference on Communication and Computer Engineering (ICOCOE), Penang, Malaysia, April 2017.
- [14] A. Ojha, R. Tiwari, K. Kadam, and K. Khot, "Web Application Development with Object Oriented Programming," 2017.
- [15] C. Ireland, D. Bowers, M. Newton, and W. Waugh, "A classification of object-relational impedance mismatch," In *Advances in Databases, Knowledge, and Data Applications, 2009. DBKDA'09, First International Conference on IEEE*, 2009, pp. 36-43.
- [16] T.H. Chen, W. Shang, Z.M. Jiang, A.E. Hassan, M. Nasser, and P. Flora, "Finding and evaluating the performance impact of redundant data access for applications that are developed using object-relational mapping frameworks," *IEEE Transactions on Software Engineering*, vol. 42:12, 2016, pp.1148-1161.
- [17] J. Kohler, C.R. Lorenz, M. Gumbel, T. Specht, and K. Simov, "A Security-By-Distribution Approach to Manage Big Data in a Federation of Untrustworthy Clouds," In *Privacy and Security Policies in Big Data*, IGI Global, 2017, pp. 92-123.
- [18] C. Bauer, G. King, and G. Gregory, *Java Persistence with Hibernate*, Second Edition, Manning Publications Co,Shelter Island, New York, 2016.
- [19] S. Sinha, *Eloquent Relations. In A beginner's guide to application development with Laravel 5.3*, Apress, 2017, pp. 123-151.
- [20] A. Alghamdi, M. Owda, and K. Crockett, "Natural Language Interface to Relational Database (NLI-RDB) Through Object Relational Mapping (ORM)," In *Advances in Computational Intelligence Systems*, Springer International Publishing, 2017, pp. 449-464.
- [21] A. Torres, R. Galante, M.S. Pimenta, and A.J. Martins, "Twenty years of object-relational mapping: A survey on patterns, solutions, and their implications on application design," *Information and Software Technology*, vol. 82, February 2017, pp. 1-8.
- [22] S. W. Ambler, *Mapping Objects to Relational Databases: O/R Mapping In Detail*, 2006.
- [23] M. Surguy. (2013, July). "History of Laravel PHP framework, Eloquence emerging". Maxoffsky. [Online]. Available: http://maxoffsky.com/codeblog/history-of-laravel-php-framework-eloquence-emerging
- [24] F.W. Davis, D.M. Stoms, J. E. Estes, J. Scepan, and M. SCOTT, J., "An information systems approach to the preservation of biological diversity. *International Journal of Geographical Information System*, vol. 4, no.1, 1990, pp.55-78.
- [25] U.S. CONGRESS OFFICE OF TECHNOLOGY ASSESSMENT, *Technologies to Maintain Biological Diversity* (Washington, D.C.: U.S. Government Printing Office), 1987.