DOMAIN ENGINEERING APPROACH FOR ANDROID CLASS LIBRARY DEVELOPMENT ON SALES AND RENTAL PROPERTY INFORMATION APPLICATIONS

Mohammad Abdul Iman Syah[1], Alif Finandhita[2]

Informatics Engineering – Universitas Komputer Indonesia
Dipatiukur Street 112-114 Bandung
E-mail : imancha_266@ymail.com[1], alif.finandhita@email.unikom.ac.id[2]

ABSTRACT

Property search trends through online property portals are increasingly prompting the development of application types of sales and rental property information. The development of features for the application of the type of sales and rental property information must be done repeatedly so that it can not minimize the time and cost required in the execution of the application. The non-availability of reusable source code is the cause of the problem. In addition, unmaintainable code resulting from the lack of analysis compared to its implementation in application development makes it difficult for developers to make improvements or development of applications built. The development of a class library is the solution to the problem. Using the FODA (Feature Oriented Domains Analysis) method, features in the applications in reusable domains are analyzed through three sample applications which are then built into a class library. Based on the results of unit testing, integration testing and user acceptance testing of the class library, it can be concluded that with the class library, time and cost in the development of the application can be minimized and can facilitate application developers in improvements or developing applications with type of sales and rental property information.

Keywwords : *Class Library, Reusable Code, Domain Engineering*

1. INTRODUCTION

The high demand for shelter and the shifting trend of society into the digital age makes online property portals increasingly popular for house hunters [1]. This triggered the development of application sales and rental property information. Development of applications with the type of sales and rental property information is still less effective and efficient. This is because there are many applications with the type of sales and rental property information but no source code that can be reused.

The unavailability of reusable source code causes in every work on the same features to be repeated over and over again so that the time and cost required in the execution of the application can not be minimized. In addition, the analysis process that is considered less compared to its implementation in every application development causes the resulting source code to be unmantainable code so that application developers find it difficult when it will do the improvements or development of applications that have been built. Based on the problem, it will be built a class library for domain application sales and rental property information on the android platform.

1.1 Sales and Rental of Properties

Property search trends through online property portals are increasing as the internet penetration is expanding today. In the midst of the lethargy of the property market last year, the search for property through the online portal observed increased. Internet penetration that now reaches 34.9% of Indonesia's population berimpak on various sectors, including real estate. When print media became a mainstay for property hunters to find the property of dreams, now slowly online property portal began to dominate [1].

Survey of real estate agents as well as manager of online property portal Lamudi revealed, 63% of Indonesia's population plans to seek shelter or move to a new home. Uniquely, they see the internet as an appropriate medium to find properties that meet their needs. The online real estate market is in fact already a favorite tool for property seekers. As many as 84.9% of survey respondents Lamudi revealed, they use online methods to find the ideal property. Only 6.7% are still relying on offline methods such as newspapers, magazines and flyers, while with other methods around 8.4% [1].

1.2 Class Library

A class library is a collection of classes created for the purpose of repeated use, either in one application or several different applications. Reuse is a basic principle in designing the program code as a class library, more specifically when the class library is open source which means the program code will also be released or presented to the consumer so that the consumer class library not only

intends to use these classes in the applications it creates, but they will also read the source code, learn it, and even improve and expand it [2].

Class library is one example of the advantages to be gained from implementing Object Oriented Programming techniques. Class libraries will speed up the construction of an app, since there is no wasted time to build or work on something that other people have made (reinvent the wheel) [2]. Testing that can be done to test a class library is to do unit testing, integration testing, and user acceptance testing.

Unit testing is commonly used to perform functions from the smallest unit of code (which can be a method, class or component) in a repeatable way. Unit testing is required to verify the code logic specific to the application. For example, on unit testing a class to ensure that the class is in the correct state. Typically, the unit code is tested separately because the test only affects and reviews changes to that unit only [3].

Integration testing is a test performed by placing related components to ensure that the modules and interfaces in an application interact with each other properly and correctly. Integration testing is done by integrating a unit with other units that interact to form a system. Integration testing is used to determine the errors and problems found in the software [4].

User acceptance testing is a formal test relating to the needs of users, requirements, and business processes undertaken to determine whether a system meets acceptance criteria or not, and other authorized users or bodies are very likely to determine whether or not to accept the system [5].

1.3 Domain Engineering

Domains are areas of knowledge restricted to maximize the satisfaction of interested parties, incorporating a set of concepts and terminologies understood by practitioners in the field, and incorporating knowledge of how to build software systems (or parts of device systems Soft) in that field [6].

The software domain is a category of every type of software available. There are seven categories of these types of software domains, including software systems, software applications, engineering / scientific software, embedded software, product-line web application software and artificial intelligence software [7].

Domain Engineering is the activity of collecting, organizing, and storing experience in building systems or parts of systems on a particular domain in the form of reusable assets ie reusable products, as well as providing Sufficient means to replicate these assets (such as qualification, dissemination, adaptation, assembly, etc.) when building new systems. Domain Engineering consists of three stages [6]:

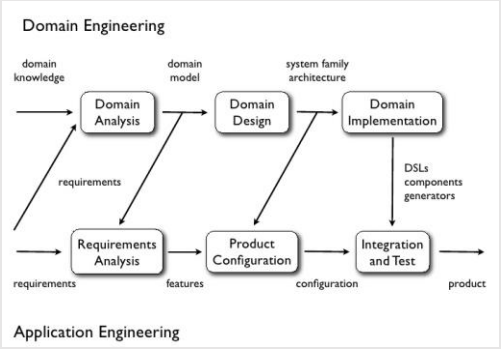


Figure 1. *Domain Engineering*

1. *Domain analysis, defines a set of needs that can be used for systems within a domain.*
2. *Domain design, building a common architecture for the system in the domain.*
3. *Domain implementation, implementing usable assets such as domain specific languages, generators, and usable infrastructure.*

1.4 Feature-Oriented Domain Analysis (FODA)

FODA is part of Model-Based Software Engineering (MBSE), a comprehensive approach to software engineering based on Domain Engineering developed by SEI where FODA is included into Domain Analysis [6].

Software Engineering Institute (SEI) introduced FODA in 1990. The methodology has further developed with subsequent usage. The FODA methodology consists of three steps: context analysis, domain modeling and modeling architecture [8].

Context analysis defines the domain scope, its relation to other domains, inputs, outputs of the domain and the needs of stored data. This is expressed as a context model and can be done using structural diagrams and context diagrams [8].

Domain modeling deals with identifying the similarities and differences of different apps within a domain. Analysis is done to get features of the domain and the standard vocabulary used for the domain. Entities and software requirements are understood using special specification techniques such as control flow and data flow. Domain modeling uses three types of techniques and produces the following output [8]:

1. *Information model, providing information in the system and the relationship between entities.*
2. *Featured model, describes the features of the system based on the end-user perspective and is useful for communicating to the user and getting a complete and consistent view of the domain. Feature models are usually represented as feature diagrams.*
3. *Operational model, functional structures and sequences that form the basis for*

developers to see how to provide features required by the user.

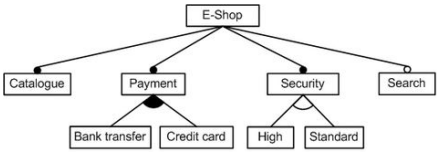


Figure 2. *Feature Diagram*

Initially, FODA has a third phase called Architecture Modeling. This phase is no longer part of FODA, but is transformed into the Domain Design phase, which follows the overall framework of Model-Based Software Engineering [6].

2. RESEARCH CONTENT

Class library that will be built is a class library for the domain of application sales and rental property information. The development of class library based on domain engineering method consists of three stages: domain analysis, domain design and domain implementation.

2.1. Domain Analysis

Domain analysis is an activity to get the similarities and differences of features that exist in the domain so as to generate a collection of reusable domain models. The applications used to obtain reusable component candidates are Lamudi, Rumah.com and Rumah123. The domain analysis based on Feature Oriented Domain Analysis (FODA) method consists of two stages, namely context analysis and domain modeling [6].

2.1.1. Context Analysis

Context analysis is the stage to determine the scope of a domain that is likely to produce a domain product. In this stage, the relationship between the domain and the external elements or entities of the domain is analyzed to know the difference as well as using the result of the context analysis as the area boundary of the domain. The result of context analysis is a context model consisting of structure diagram and context diagram [6].

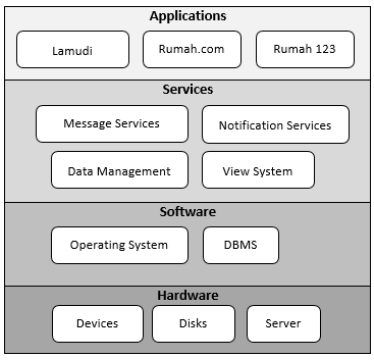


Figure 3. *Structure Diagram*

The picture above shows the structure diagram of the application in the domain. Based on the image, there are four layers that exist in the application in the domain of Applications, Services, Software and Hardware.

Layer applications is a layer that contains a collection of applications with the type of sales and rental property information used as a sample to get the library that can be reused. There are three sample applications on the application layer used, which consists of applications Lamudi, Rumah.com and Rumah123.

Layer services is a layer that contains a collection of services provided by applications that exist on the application layer consisting of four services:

1. *Data Management, a service used for data management.*
2. *Message Services, a service that is used for sending messages in the form of text messages, email and phone.*
3. *Notification Services, a service used for notification delivery to users.*
4. *View System, which is a service used to display the interfaces that exist in each application.*

Layer software is a supporting layer in the form of software used by applications in a domain consisting of two devices:

1. *Operating System, which is software required by apps in the domain to run the services provided.*
2. *DBMS (Database Management System), which is software needed to store data.*

Layer hardware is a supporting layer in the form of hardware used by applications in a domain consisting of three devices, namely:

1. *Devices, which are hardware required by apps in the domain to display the services provided.*
2. *Disks, which is the hardware needed to store the data used.*
3. *Server, the hardware required by the application so that the data stored can be accessed through a network.*

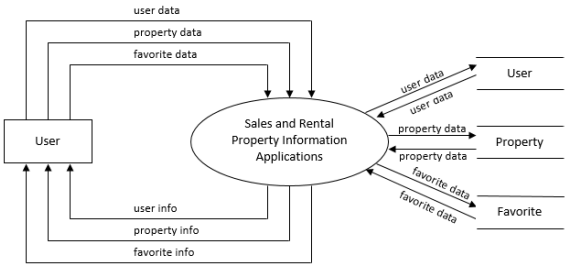


Figure 4. *Context Diagram*

The picture above shows the context diagram of the application in the domain. Based on the picture, there is one entity that interact directly with the application in the domain ie User. In addition, there

are three data stores that exist in the application in the domain that is User, Property and Favorite.

2.1.2. Domain Modeling

Domain modeling is a step to identify and model the similarities and differences of features that exist in each application in the domain based on the results of the context analysis as the limit. Domain modeling consists of three stages, namely information analysis, feature analysis and operational analysis [6].

2.1.2.1. Information Analysis

Information analysis aims to describe domain knowledge in the form of entities and relationships among entities on the domain. The result of information analysis is information model which can be semantic network, entity-relationship model or object-oriented modeling [6].

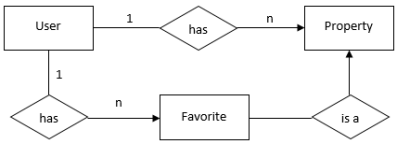


Figure 5. *Entity-relationship Model*

The picture above shows the entity-relationship of the application in the domain. Based on the picture, there are three entities in the application in the domain that is User, Property and Favorite.

2.1.2.2. Feature Analysis

Feature analysis is a step to know the features provided in each application in the domain from the end-user perspective. The similarity and feature differences of each application are also analyzed. The result of feature analysis is a feature model [6].

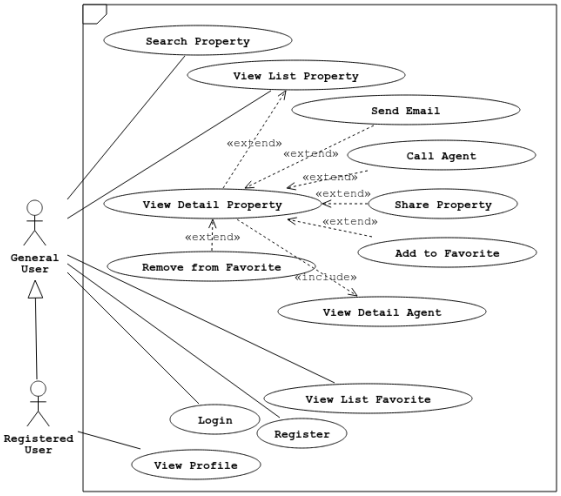


Figure 6. *Use Case Lamudi*

The picture above shows the use case of Lamudi application. Based on these images, there are two user levels of common users and registered users and there are thirteen use cases in Lamudi applications.

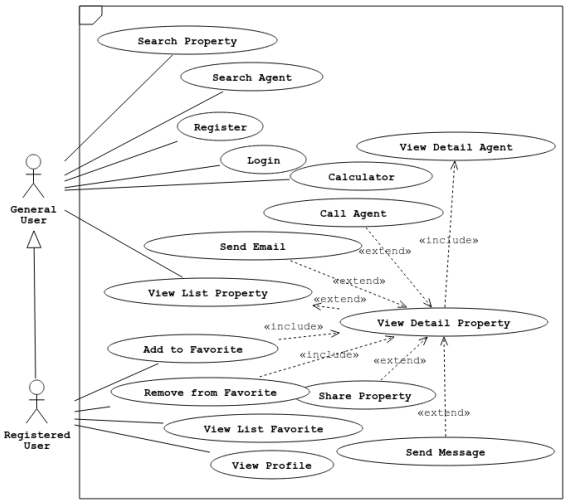


Figure 7. *Use Case Rumah.com*

The picture above shows the use case of Home.com application. Based on these images, there are two user levels of common users and registered users and there are sixteen use cases that exist in the application Rumah.com.

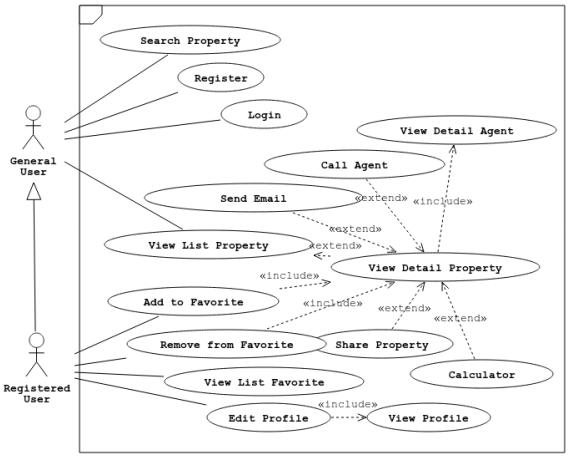


Figure 8. *Use Case Rumah123*

The picture above shows the use case of the Rumah123 application. Based on these images, there are two user levels of common users and registered users and there are fifteen use cases in Rumah123 applications.

Based on the analysis of features of each of the above applications, then performed an analysis of the similarities and differences of features that exist in each application in the domain.

Table 1. *Commonality Feature*

Feature	***Lamudi***	***Rumah .com***	***Rumah 123***
Search Property	√	√	√
Search Agent	-	√	-
View List	√	√	√

Feature	Lamudi	Rumah .com	Rumah 123
Property			
View Detail Property	√	√	√
Send Email	√	√	√
Send Message	-	√	-
Call Agent	√	√	√
Share Property	√	√	√
Add to Favorite	√	√	√
Delete from Favorite	√	√	√
View Detail Agent	√	√	√
Calculator	-	√	√
View List Favorite	√	√	√
Register	√	√	√
Login	√	√	√
View Profile	√	√	√
Edit Profile	-	-	√

The table above shows the similarities and differences of features in each application. Based on the above table, there are four features that are only owned by one application, there is one feature that is not owned by one of the applications and there are fifteen features owned by each application.

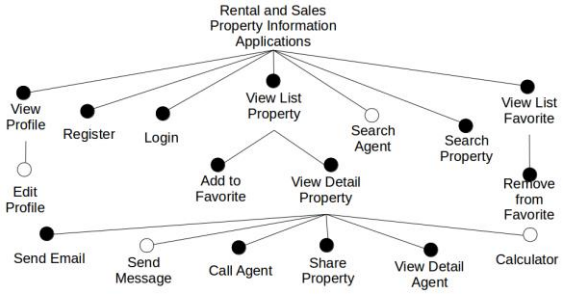


Figure 9. *Feature Diagram*

The picture above shows the feature diagram of the application in the domain. Based on the above diagram feature, there are two types of features available in the application in the domain of mandatory features and optional features. The features that will be used as a class library are all mandatory features and optional features that may be required in the construction of applications in the domain.

Table 2. *Class Library Feature*

Feature
Search Property
View List Property
View Detail Property
Send Email
Call Agent
Share Property
Add to Favorite
Delete from Favorite
View Detail Agent
Calculator
View List Favorite
Register
Login
View Profile

The table above shows the features that will be used as a class library. Based on the table, there are fourteen features that will be used as a class library.

2.1.2.3. Operational Analysis

Operational analysis is a step to describe control and data-flow in the application in the domain based on context analysis, information model and feature model that has been done. The result of operational analysis is operational model [6].

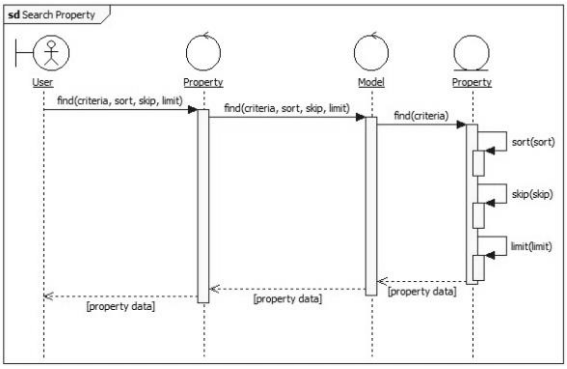


Figure 10. *Sequence Diagram Search Property*

The picture above shows the operational model of the property search feature. Based on the picture there are two classes of Property and Model and one entity is Property used for property search feature.

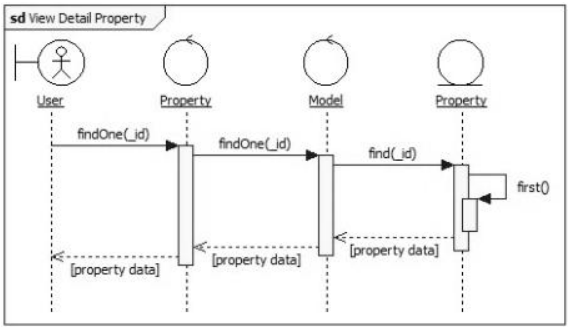
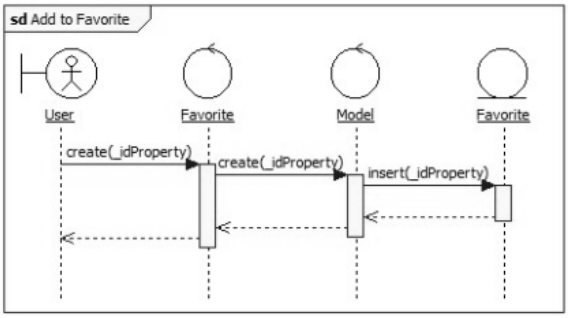


Figure 11. *Sequence Diagram View Detail Property*

The picture above shows the operational model of the feature view detail property. Based on the picture there are two classes of Property and Model and one entity is Property that is used for view detail property feature.



Gambar 12. *Sequence Diagram Add to Favorite*

The picture above shows the operational model of the added feature to the favorites. Based on the picture there are two classes namely Favorite and Model and one entity Favorite is used for add to favorite feature.

2.2. Domain Design

Domain design is the design stage of the class library to be built which consists of class and package design.

2.2.1. Class Design

Based on the results of the analyzes that have been done, then the features that will be used as class libraries are then categorized and / or classified by context, linkage of functions and data it processed to determine the name of the class and its method.

Table 3. *Feature Classification*

Feature	***Class***
Add to Favorite	*Favorite*
Delete from Favorite	
View Detail Favorite	
Search Propertu	*Property*
View List Property	
View Detail Property	
Share Properti	
Calculator	
View Detail Agent	*User*
Send Email	
Call Agent	
View Profile	
Register	
Login	
-	*Model*
-	*Database*

The table above shows the categorization of features for the class library. Based on the table, there are five classes that will be built namely Favorite, Property, User, Model and Database class.

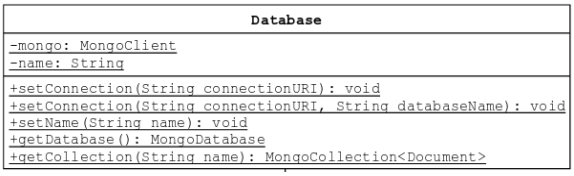


Figure 13. *Class Database*

The picture above shows the Database class structure in the class library. Based on the picture there are two attributes of mongo and name and five methods are setConnection, setConnection, setName, getDatabase and getCollection that exist in the Database class.

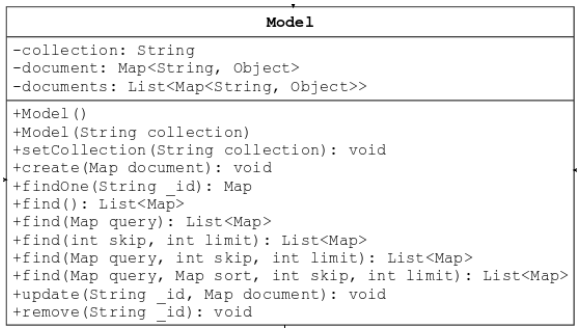


Figure 14. *Class Model*

The picture above shows the class structure of the Model in the class library. Based on the picture there are three attributes that are collection, document and documents and ten methods are setCollection, create, findOne, find, find, find, find, find, update and remove existing in the Model class.

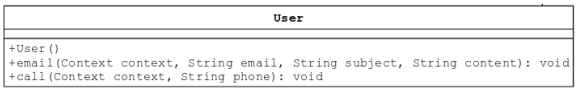


Figure 15. *Class User*

The picture above shows the class structure of the User in the class library. Based on the picture there are two methods of email and call that exist in the User class.

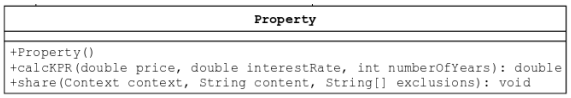


Figure 16. *Class Property*

The picture above shows the structure of the Property class in the class library. Based on the picture there are two methods of calcKPR and share that exist in the Property class.

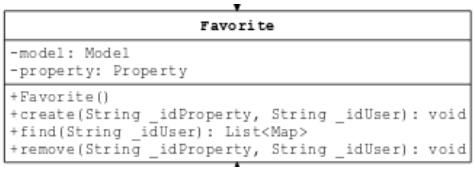


Figure 17. *Class Favorite*

The picture above shows the Favorite class structure in the class library. Based on the picture there are two attributes that are model and property and three methods are create, find and remove that exist in Favorite class.

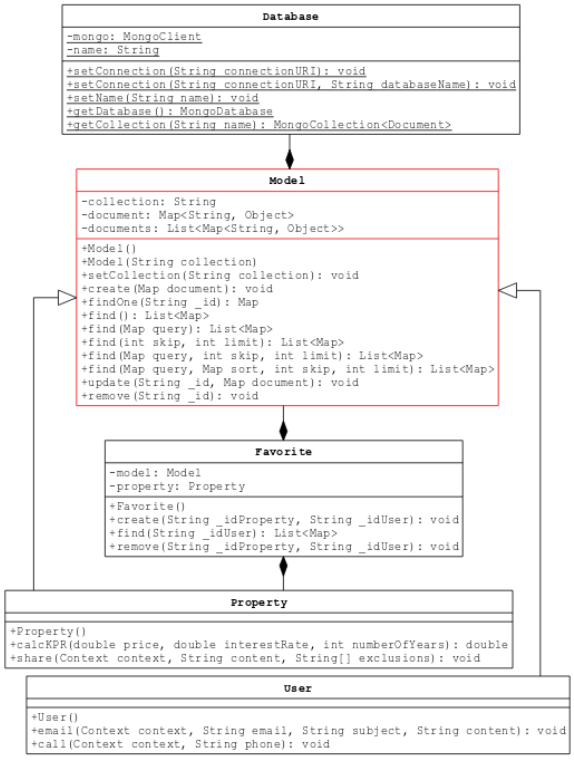


Figure 18. *Class Diagram*

The picture above shows the relationship or relationships between classes that exist in the built library class. The Class Model has a composition relation with the Database class. The Favorite class has a composition relation with the Model and Property class. Class Property and User class have a generalization relationship with the Model class. Class Model is a class that should not be accessed directly in its use, but must go through the derived class.

2.2.2. Package Design

Package design is the stage to design the package based on the design of the classes that have been done. Package design aims to facilitate in understanding and calling from the built class structure.

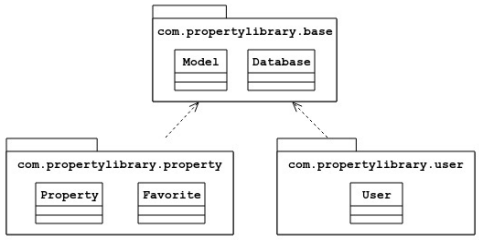


Figure 19. *Package Design*

Based on the picture, there are three packages in the built library class: com.propertylibrary.base, com.propertylibrary.property and com.propertylibrary.user. In the package com.propertylibrary.base there are classes Model and Database. In package com.propertylibrary.property there is class Property and Favorite. While on package com.propertylibrary.user there is class User. Package com.propertylibrary, property and com.propertylibrary.user have depencency against package com.propertylibrary.base.

2.3. Domain Implementation

Domain implementation is the stage of development of the class library to be built by creating packages and classes in accordance with the design that has been done. It will then integrate the packages and classes that have been created into a class library and implement and test the class library using unit testing, integration testing and user acceptance testing.

2.3.1. Class Library Implementation

Implementation of class library is a step to implement from class library built. Class libraries that have been built are compiled into android archive file form (.aar) for easy use.

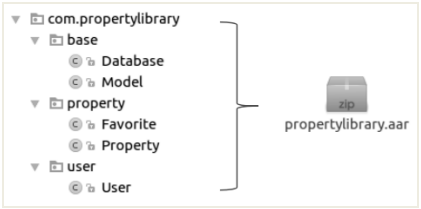


Figure 20. *Class Library*

The above image is the result of a compilation of the built-in library class. Implementation of the class library is by using the class library in the construction of information applications sales and rental property. The steps are as follows:

1. *Download the class library via maven repository [9]*

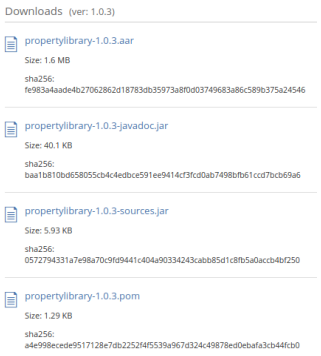


Figure 21. *Download the Class Library*

2. *Add the class library as the dependency of the project*

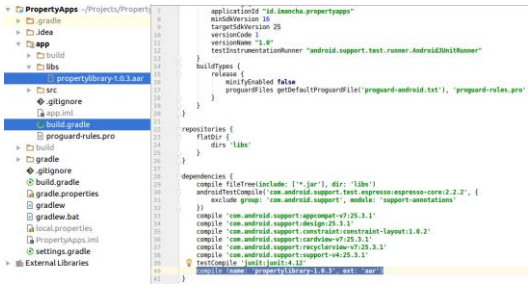


Figure 22. *Add the Class Library*

3. *Using class and method as needed*

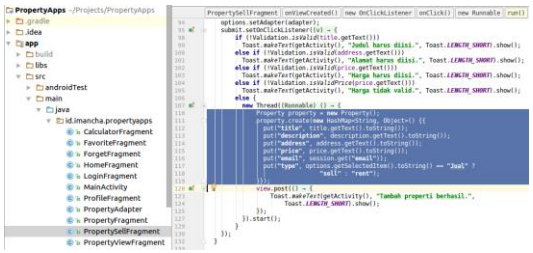


Figure 23. *Use the Class Library*

2.3.2. Class Library Testing

Testing the class library is a step to test the class library that has been built to ensure that the built class library has been in accordance with the goals and needs. Class library testing consists of three types: unit testing, integration testing and user acceptance testing.

2.3.3.1. Unit Testing

Unit testing is a step to test each unit in the form of methods that exist in the class library that has been built. Unit testing aims to ensure that all methods that have been made in accordance with its function.

Table 4. *Unit Testing*

Test Component (Class)	***Test Unit (Method)***	***Test Type***	***Result***
Database	*setConnection(String connectionURI)*	*Unit Testing*	*Pass*
	setConnection(String connectionURI, String databaseName)	*Unit Testing*	*Pass*
	setName(String name)	*Unit Testing*	*Pass*
	getDatabase()	*Unit Testing*	*Pass*
	getCollection(String name)	*Unit Testing*	*Pass*
Model	*setCollection(String collection)*	*Unit Testing*	*Pass*
	create(Map document)	*Unit Testing*	*Pass*
	findOne(String _id)	*Unit Testing*	*Pass*
	find()	*Unit Testing*	*Pass*
	find(Map query)	*Unit Testing*	*Pass*
	find(int skip, int limit)	*Unit Testing*	*Pass*
	find(Map query, int skip, int limit)	*Unit Testing*	*Pass*
	find(Map query, Map sort, int skip, int limit)	*Unit Testing*	*Pass*
	update(String _id, Map document)	*Unit Testing*	*Pass*
	remove(String _id)	*Unit Testing*	*Pass*
Property	*calcKPR(double price, doubleinterest Rate, int numberOfYears)*	*Unit Testing*	*Pass*
	share(Context context, String content,	*Unit Testing*	*Pass*

Test Component (Class)	Test Unit (Method)	Test Type	Result
	String[] exclusions)		
Favorite	create(String _idProperty, String _idUser)	Unit Testing	Pass
	find(String _idUser)	Unit Testing	Pass
	remove(String _idProperty, String _idUser)	Unit Testing	Pass
User	email(Context context, String email, String subject, String content)	Unit Testing	Pass
	call(Context context, String phone)	Unit Testing	Pass

The table above is the scenario and the result of unit testing in the class library. Based on the table it can be concluded that each unit in the form of methods on the built class library is running as expected both in terms of validation and error handling process and produce the desired output.

2.3.3.2. Integration Testing

Integration testing is a testing phase by combining several units that exist in the class library. Integration testing aims to ensure that each class unit that has been built can be integrated or used with other units such as on the page interface. Integration testing is done by building an application using the built-in class library.

Table 5. *Integration Testing*

Test Component (Class)	Test Unit (Feature)	Test Type	Result
Property	Add Property	Integration Testing	Pass
	Search Property	Integration Testing	Pass
	View Property List	Integration Testing	Pass
	View Detail Property	Integration Testing	Pass
	Share	Integration	Pass
	Property	Testing	
	Calculator	Integration Testing	Pass
Favorite	Add to Favorite	Integration Testing	Pass
	Delete from Favorite	Integration Testing	Pass
	View List Favorite	Integration Testing	Pass
User	View Detail Agent	Integration Testing	Pass
	Send Email	Integration Testing	Pass
	Call Agent	Integration Testing	Pass
	View Profile	Integration Testing	Pass
	Register	Integration Testing	Pass
	Login	Integration Testing	Pass

The table above is the scenario and the result of integration testing in the class library. Based on the table it can be concluded that each unit class library built can integrate with other units into a component and function in accordance with the desired features.

2.3.3.3. User Acceptance Testing

User acceptance testing is a test performed in the user environment where users perform an assessment of the class library that was built using the media interview.

Table 6. *User Acceptance Testing*

Question	Answer
Is the class library can help in building the type of application sales and rental property information on the android platform?	Very helpful because if there is no class library every want to make application sales and rental property information on android device should start from the beginning.
Is the class library meets the general functionality of the type of sales and rental property information applications on the android platform?	Overall it has been fulfilled to achieve minimum viable product.
Is the class library easy to use?	Yes, but it may take some time to adjust to the

Question	*Answer*
	schema in the class library.
Will the class library make source code more structured?	*With the class library, the source code can be made more structured because the built in class library has been created in accordance with its class features.*
Is the class library can minimizes the time and cost of the application work?	*When compared to the work of features using the class library and without using the class library then can be spelled out if the use of features using the class library faster and can save time and cost.*
What are the lacks of the class library?	*Class library that is built is still minimal validation and error handling on the method, although there are already some. So also for the documentation may be more in clarify again.*

The above table is the scenario and the result of user acceptance testing. Based on the table it can be concluded that the built class library has been in accordance with the goal, which is easier for application developers in building or developing the type of application sales and rental property information on the android platform.

3. CONCLUSION

Based on the results that has been submitted from the analysis, implementation and testing on the class library can be concluded that:

1. *Class library can minimize the time and cost of building applications with domain sales and rental property information.*
2. *Class library can facilitate application developers in improving or developing applications with the domain of sales and rental property information.*

Suggestions for the development of the next class library is:

1. *Added functionality that does not exist in the built-in library class like functionality for transactions.*
2. *Complete the class library usage documentation so that application developers have no trouble using the class library.*
3. *Adding more validation and error handling to each method in the class library makes it easy for app developers to find out the mistakes.*

REFERENCES

[1] Properti, *Tren Pemasaran Properti Dalam Jaringan Kian Populer & Dominan*, 2016. [*Online*]. *Available*: http://properti.bisnis.com/read/20160210/107/517761/tren-pemasaran-properti-dalam-jaringan-kian-populer-dominan. [*Accessed*: 14- Feb- 2016].

[2] M. Azis, *Object Oriented Programming dengan Php 5*. Jakarta: Elex Media Komputindo, 2005.

[3] G. Testing, *Getting Started with Testing | Android Developers*, Developer.android.com, 2017. [*Online*]. *Available*: https://developer.android.com/training/testing/start/index.html. [*Accessed*: 14- Feb- 2016].

[4] J. Hermawan, Analisa Desain & Pemrograman Berorientasi Obyek dengan UML dan Visual Basic.NET. Yogyakarta: Andi, 2004.

[5] M. Huda and B. Komputer, *Membuat Aplikasi Database dengan Java, MySQL, dan NetBeans*. Jakarta: PT Elex Media Komputindo, 2010.

[6] Czarnceki, Krzysztof, Ulrich W. Eisenecker, *Generative Programming: Methods, Tools, and Applications*, Addison-Wesley, 2010.

[7] R. Pressman, *Software engineering: a practitioner's approach, 7th ed*., McGraw-Hill, 2010.

[8] Alana E., Rodriguez A., GMV Domain Engineering Methodology Survey. CORDET, 2007.

[9] propertylibrary, Bintray.com, 2017. [Online]. Available: https://bintray.com/imancha/maven/propertylibrary. [Accessed: 10-Aug-2017].