

Study and Implementation Information System of Zakat using MVC Architecture

M A Akbar¹, I Handriani¹

¹Information System, University of Mercubuana, Indonesia

¹E-mail: ali.akbar@mercubuana.ac.id, inge.handriani@mercubuana.ac.id

Abstract. Alm's Giving(Zakat) is one from ve obligations of muslim. Many muslim known that regulated as well, especially for ZAkut Fitrah and ZAkut Maal. This ritual usually performed in approaching moment Idul Fitri. For many years, there are was always prob lems on those season.To overcome the problem, the use of Information system could be considered as solution, it makes ZAkut handout and distribution become organized and transparent. This paper studied the Implementation Information System of ZAkut with exible component. The combination of the technology Struts 2 Framework and hibernate was implemented to the system. This system is provided with the advantages of Struts2 MVC characterized to expansibility and maintainability.

1. Introduction

There are five primary obligations of a muslim or better known as the pillars of faith, that each Muslim must ful ll in their lifetime. Zakat or alms giving is the payment from a prescribed proportion of a quali ed Muslim's income or possessions for the welfare of the community particularly the neediest members of the Muslim community[1].

In practice, there are several problems associated with the collection, validation, and distribution of Zakat[2][3][4]. The major problem is due to the disribution of zakat which is still under expectations. In 2015, the Zakat collection is nearly 3,653 Billion IDR, however the distribution only 2,251 Billion IDR. The gap between colleting and distributing is nearly about 40[4].

The second problem, in every district and villages the number of poor families is di erent. Zakat should be given to those who do need and classi ed as recipient(Mustahiq). Many agencies perform eld survey and interviews to determine and registered recipient and deciding which candidate can recieve the Zakat. This problem will be simpli ed if every village mosque entity performs the task.

The third problem is the lack of public interest in zakat is due to the prevalence in the community that zakat is the annual moment before Idul Fitri holidays for all items of zakat, and the lack of public trust in management and zakat management risk. The problems that arise in the community in the management of zakat are:

- Communities or Muslims tend to pay zakat to mosques around their homes rather than to official institutions like the Badan Amil Zakat (BAZ) and the Lembaga Amil Zakat (LAZ).
- BAZ and LAZ have also cooperated with district o ce and region in Jakarta to optimize zakat management through mosques around the district.
- Lack of education related to zakat along with items of zakat.
- Issue about profesionalism of the committee of zakat, usually the zakat committee is the committee of the annual moment ahead of Idul Fitri, and dissolved after Idul Fitri.

From the above explanation about the management and risk of zakat, and also the problems that arise in the community can be concluded to optimize the management of zakat required a User Friendly Information System that can be useful. In this study, we propose a system design of Zakat information system which used designated Mosque board to solve the annual moment Idul Fitri.



2. Concept and Related Work of Zakat

Zakat is one of the islamic religious buildings, zakat it's self according to the terms is a mandatory form of worship for every muslim. Zakat is worship by removing some of the possessions owned to be channeled to those who are entitled to receive it. In general, zakat itself is a form of social solidarity that can strengthen the ropes of brotherhood between people and nation. Zakat can also strengthen the country's economic resilience because it can become a bridge between the strong economy and the weak economy.

There have many studies work on system implementation of zakat. Some of them perform the web-based application system for data entry Muzakki and Mustahiq with ease and have an organized database. Gufroni et.al doing the development web-based application tries to regulate and connect between the process and management of Zakat based on Mosque on Tasikmalaya City.[5] Eliyani and Inge perform the study the Zakat acceptance model conducted by national level is as good as in the central. The study was carried out on the Zakat Management Organization managed by government and the community. The study was conducted within 1 year intervals, from January- December 2013.[6] Herdyansyah perform the study about vulnarable and losing data if Region Amil Zakat Board (BAZDA) manually entering the data Mustahiq and Muzakki.They designed a Web-based application system for data entry Muzaki and Mustahiq with ease and have an organized database so that the data nedded more easily inputted. They use PHP for Web-based and use database MySql [7] .

Maulana et.al conduct research determining Mustahiq based on the priority using weight product method. The criteria of Mustahiq selection that used on they application are monthly income, home, vehicle ownership, and also the number dependents, they using data from BAZIS-Jakarta[8].

Subri on his thesis has developed a decision support system which simulate Mustahiq determination using Analytical Hierarchy Process(AHP)[9]. Therefore, in this study we propose a design Information System of Zakat using MVC architecture based on J2EE Framework Struts 2, Hibernate and Oracle 10g for database.

3. MVC Architecture

MVC pattern based on J2EE arhitecture has been applied in many elds, but there is a lot of complexity. Some of the technical personnel at home and abroad use di erent methods to study it from the di erent aspect[10][11][12]. Many Web project development nowaday use this design pattern. MVC due to Model; View and Controller which in practice separating business logic, data presentation and applied bahavior. The model which represents business data and business logic is main part of the application program. View is the interface through which users can access and query the state of the business. View also can accept model requirements to update data andd renew user interface synchronously. The role of controller includes accepting the request from client, choosing appropriate business logic to implement and then sending the results back to client [13].

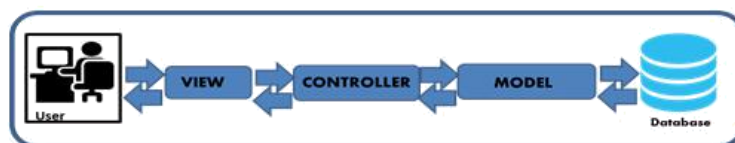


Figure 1. MVC Architecture

3.1. Struts 2

The Struts 2 adopts the MVC design pattern and is an improvement of Struts 1 and Web Work. Struts framework has realized a good software design idea in a practical and expensible way, and has advantages of modularization, exibility and reuse of components[14].

In its design Model is used to implement business logic and usually achieved by JavaBeans or Plain Old Java Object (POJO), View is used to interact with users usually achieved by JSP pages, and Controller is used to implement business logic process control mainly achieved by the core controller of Struts 2 FilterDispatcher and Action [14].

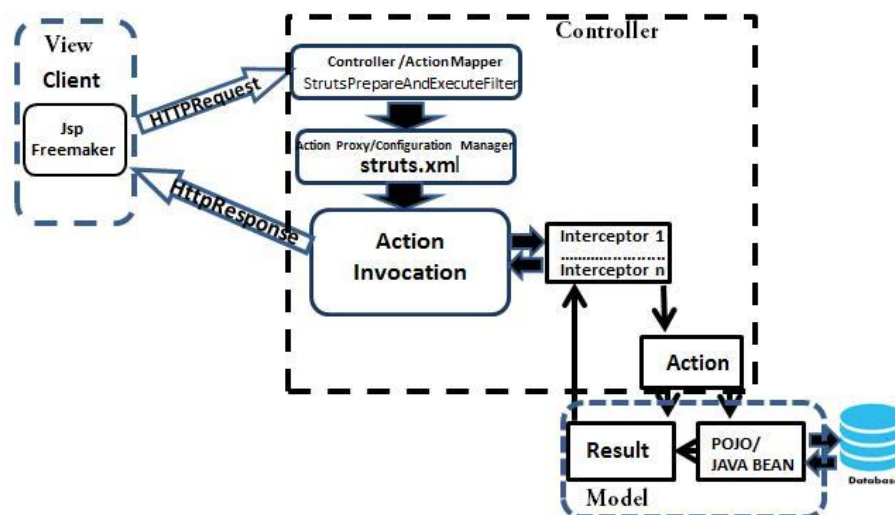


Figure 2. Struts 2 Architecture

Fig 2 shows the work ow of the architecture from the Struts 2 Framework. Interaction with the user will be forwarded in the form of a request as HttpRequest, then the Struts 2 core controller will forward the request and read on the struts.xml form. The core controller interceptor will intercept the client request then it will set the request to appropriate action object according to struts.xml. In the action object occasionally a request that only runs a function without accessing the data object, most of which is interacting with data object in the databases. The result of this data processing will be transmitted back to the interceptor, and the interceptor will produce a responses that matches with the return value on the JSP pages.

3.1.1. struts.xml here the struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC .....
"classpath://struts-2.0.dtd">

<struts>
  <constant name="struts.enable.
DynamicMethodInvocation" value="false" />
  <constant name="struts.devMode"
value="false" />
  <constant name="struts.custom.
i18n.resources"
value="ApplicationResources" />
  <package name="sizaki"
extends="struts-default" namespace="/">
    <interceptors>
      <interceptor name="timerInterceptor"
class="sizaki.interceptor.
TimerInterceptor">
        <interceptor-ref name="defaultStack" />
        <interceptor-ref name="execAndWait">
```

```

</interceptor>
<interceptor name="validation"
class="org.apache.struts2.interceptor.
validation.
AnnotationValidationInterceptor"/>
<interceptor-stack name="timerStack"> </interceptors> <interceptor-ref
name="timerInterceptor" /*/> Class Action */

<param name="excludeMethods">*</param>
</interceptor-ref>
<interceptor-ref name="defaultStack" /> </package>
</interceptor-stack>
<interceptor-stack
name="executeAndWaitStack">
.....
.....
.....
</struts>

```

3.2. Hibernate

Hibernate is an Open Source Object-Relational Mapping(ORM) framework product of Apache Software Foundation[13]. Hibernate is a bridge between Java application and database, mapping between Java Objects and database table are created by writing the object-relational mapping les, and it provides a lot of methods that encapsulate data access operations to greatly reduce the time manual writing SQL and JDBC to process data access[14].

Hibernate has a lightweight object encapsulation for JDBC to reduce the complexity of the persistence layer, which makes developers pay more attention to the business logic of the application system rather than the underlying data structure [14].

Hibernate has five core interfaces, they are Session, Session Factory, Transaction, Query and Configuration. Programmers can access persistent objects and also can control transactions with these interfaces. Systems built by Hibernate can avoid manually developing the methods of the data persistence layer. It will simplify the development difficulty and accelerate development speed[15].

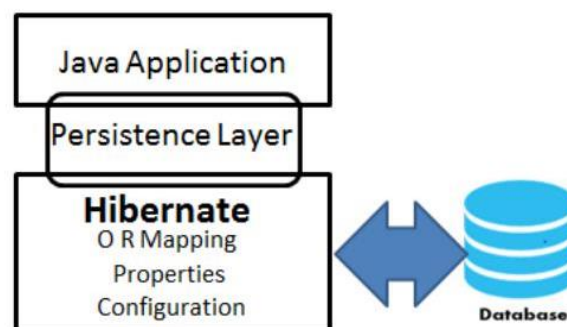


Figure 3. Hibernate Architecture

3.3. System Overall

Fig 4 shows the system overall, the image shows the result of integration between Struts2 framework with hibernate ORM application. It is clear that on the Struts 2 architecture the action option will access java bean / POJO, with integration on hibernate the action option will access the layer on hibernate. The object from action will transmit to the persistence layer, then through to Object Relation Mapping; configuration; then they will access the data via Hibernate schema. The results of the processed data will be returned as the result value according to the architectural scheme in the picture above.

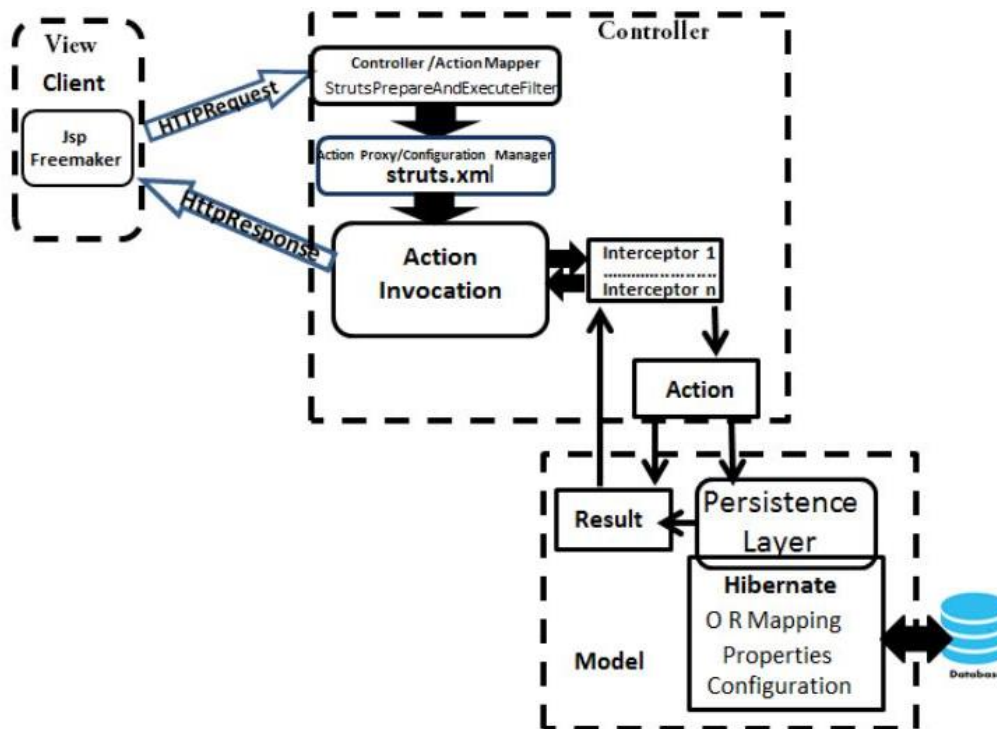


Figure 4. Architecture Overall

The advantage from this architecture is a capability to reduce overload in accessing database with smart fetching strategies. It means, if there are two or more users accessing the same table from the same scheme then hibernate will be the layer that can manage and also set the data object queue to access the database. so the execution queue is on the side of the hibernate layer rather than on the database side, on the database side looks like a semi-single execution object. The problems from users accessing the same table from the same scheme, this is not an issue if it applies to the number of user approaching 100 users, but this scheme will be an issue if this applied on the execution of more than 1000 users. Databases tend to be deadlocks, a serious implication in the event of deadlocks is to restart the server databases. This becomes very serious because the business process will be disrupted and the consequences of disruption of the system is a loss on the transaction.

4. Implementation System Design

The system consisted of user management, Mustahiq management, Muzaki management, Zakat transaction, quality of zakat item, and periode time of zakat. Because of limitation length, the paper takes user management for example to describe implementation MVC in process of Zakat.

In the implementation system, Struts2 is used for the presentation layer to provide clear business processes and user interaction and achieve loose decoupling of the presentation layer and logic layer, besides it would get rid of the high coupling of original development pattern[?]. The sequence diagram of application is shown in Fig- 5.

Fig-5 shows the sequence diagram, the diagram shows the process of user interaction in the system, in this case that is discussed is the link scheme login with the MVC process. After user submit, the system will invoke access via controller (business logic).

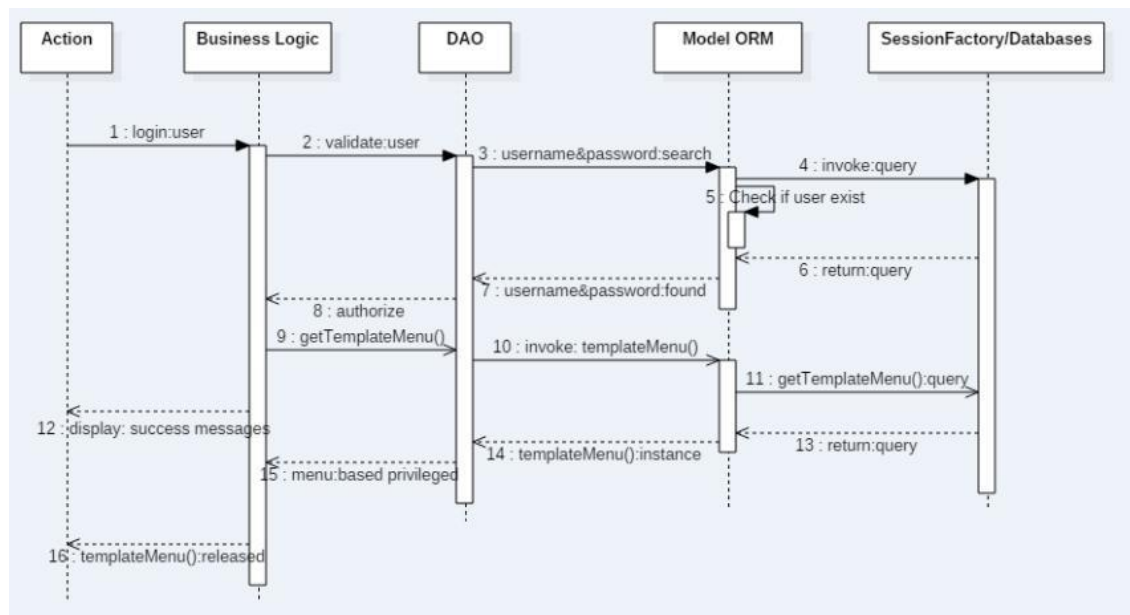


Figure 5. Sequence Diagram Login Schema

The controller checks the user's validation and performs the query through the DAO class. When DAO class perform query, the system will invoke session via SessionFactory.

```

hsession = (Session) HibernateUtil.getSession();
StringBuilder qryStr = new
    StringBuilder();
qryStr.append("select username, password from masteruser ");
qryStr.append("where username =:UserName and password =:passWord ");
SQLQuery q = hsession.createSQLQuery(qryStr.toString());
q.setString("UserName", userName);

```

```

q.setString("passWord", userpass);
List<Object[]> result = q.list();

```

In this section, the Hibernate util is a substitute part of Connection in JDBC part. The advantages is we do not have to make the Connection Class if we want make interaction with databases. SessionFactory interface in Hibernate lets us create session and reuse the session. Fig-6 shows the prototype of the login view was created.

```

public class HibernateUtil {
    private static final Logger logger
    =Logger.getLogger(HibernateUtil.class);

    public static Session getSession() {
        boolean isError = false;

        int errorCode = 0;

        Session session = null;

        do {

            try {
                session =

                errorCode = ex.getErrorCode();

                try {
                    Thread.sleep(10000);
                } catch (InterruptedException ex1) {

                    logger.error("Sleep Waiting
                    recreate Session fail");

                    logger.error(ex1, ex1);

                }}}

                while ((isError) &&(Arrays.asList(errCodes)

```

```

        .contains(errorCode)));
getSessionFactory().openSession();
ThreadLocalSessionContext.bind(session); return session;
    }

testSession(session);

logger.debug("Session Valid");

isError = false;

} catch (SQLException ex) {

logger.error("Error Test Session");

logger.error("Error Code : "
+ ex.getErrorCode());

logger.error(ex, ex);

isError = true;

    public static void
    closeSession(Session session) {
        if (session != null) {
            ThreadLocalSessionContext.
            unbind(getSessionFactory());
            session.close();
        }
    }
}

```



Figure 6. Login Schema

5. Conclusion and Future Work

This paper describe the design and prototype system for Zakat information system based on MVC architecture. The implementation was build with Struts framework and hibernate technology. This system provides the options for process and zakat management. Because many zakat application was developed in the php platform. the advantages of this java platform architecture are implemented in the zakat program.

The system show us that the technology provides code separation, development cycle efficient and effective, also enhances system maintainability and reusability. The System also can provide a

variety of user access in accordance with its role in the application. Besides, in terms of zakat implementation. This research tries to solve some zakat constraint especially try to minimizing the gap between distribution and acceptance of zakat.

The results obtained from this research is the implementation of MVC architecture in gure-4 in terms of process and management of Zakat. The great expectation by using this architecture can solve the problem annual moment Idul Fitri, it makes Zakat handout and distribution become organized and transparent. Considering at the moment there is a surge transactions. This architecture is widely used in many information systems applications in banks, nance and insurance companies that have large transactions at a time.

In the future, we will make the system of handout and distribution of zakat based on the architecture. A map based system that can map who is eligible to receive zakat, and the zakat can choose from the system of who is eligible to receive zakat. Muzakki can choose from the system whom the person is eligible to receive.

References

- [1] Hidayah Sulaiman and Norziana Jamil. 2014. "Information Security Governance model to enhance zakat information management in Malaysian Zakat Institutions ". International Conference on Information technology and Multimedia IEEE. pp 200-205
- [2] F. Rahmawati, A.R. Hidayat, and T.Suprihatin. "Analisis Pengelolaan Zakat dalam Upaya Meningkatkan Kesejahteraan Mustahiq di Badan Amil Zakat Nasional Kota Cimahi".in Keuangan dan Perbankan Syariah. pp 503-508
- [3] A.S.Jahar. 2010."Masa Depan Filantropi Islam Indonesia Kajian Lembaga-lembaga Zakat dan Wakaf". in Annual Conference on Islamic Studies
- [4] H. T. Sukmana. et.al. 2017. "The Prototype of ZAKat Management System in Indonesia by Using the Social Society Approach: A Case Study". 5th International Conference on Cyber and IT Service Management (CITSM). pp 1-4
- [5] A.I.Gufroni, I. Wisandani, and H. Sukmawati. 2014. "Sistem Informasi unit Pengumpul Zakat Terintegrasi (Studi Kasus BAZNAS kota Tasikmalaya)". Vol.3, No.4, pp 236-241.
- [6] Eliyani and Inge Handriani. 2016. "Zakat Acceptance System on National Level of Zakat Management Organization in Indonesia". Journal of Islamic Financial Studies. Vol.2.No.2 pp 67-74
- [7] E.Herdiyansyah et al. 2013. "Pengembangan Aplikasi Pendaftaran Muzakki dan Mustahik berbasis WEB di Badan Amil Zakat Daerah Kabupaten Garut". in Algoritma.Vol:10 STT Garut. pp. 1-10.
- [8] H.Okta Maulana, et al. 2017. "Application for Determining Mustahiq Based on Priority using Weight Product Method(Case Study: BAZIS DKI Jakarta)". in 5th International Conference on Cyber and IT Service Management (CITSM). pp. 1-4
- [9] A. Subri. 2011."Rancang Bangun Sistem Penunjang Keputusan Penentuan Mustahik dengan Pendekatan Analytical Hierarchy Process (AHP)". Thesis
- [10] Fang Li-Xia. 2015."English Course Online Learning System Design Based on MVC". International Conference on Intelligent System Design and Engineering Application
- [11] N.Bessus, P. Norrington, F. Xhava. 2011. "Technology Enchanced Systems and Tools for Collaborative Learning Sca olding". Vol.350. pp-1-25
- [12] A.Juan, Th Daradoumis, J. Faulin and F. Xhafa. 2009."SAMOS: A Model for monitoring Students and Group Activity in Collaborative e-Learning".International J Learning Technology . pp 200-205
- [13] Erxiang Chen and Minghui Liu. 2009."Research Design on Library Management System Based on Struts and Hibernate Framework". WASE International Conference on Information Engineering. pp.310-313

- [14] Wang Yun and LIU Xiao-Xia. 2009. "Design and Implementation of Web Application Framework Based on S2SH". fourth International Conference on Multimedia Information Networking and Security. pp. 117-120
- [15] Dong Mei, Zhang Huanshui. 2007. "Implementation of Funds Management Information System Based on Struts and Hibernate". Journal of Wuhan University of Technology (Transportation Science & Engineering). Vol.31, pp.1106-1109