ISSN 2355-0082

E-Government Integration of Sidoarjo Regency using Service Oriented Architecture (SOA)

Adi Putra Utama[1], Rengga Asmara[2], Jauari Akhmad Nur Hasim[3]
Department of Informatics and Computer Engineering
Electronic Engineering Polytechnic Institute of Surabaya
Surabaya, Indonesia
[1]adiputrautama17@gmail.com, [2]rengga@pens.ac.id, [3]jauari@pens.ac.id

Accepted on 6 November 2019
Approved on 20 December 2019

***Abstract*— the slow process of government services is a major problem of current government service systems (e-government). This is due to the lack of integration between e-government systems with regional governments. The application of data integration can accelerate the process of interaction and communication between agencies in the regional government. This study aims to develop an integrated e-government from various services that exist in government services in Sidoarjo Regency. The architecture used in developing this integration is using the Service Oriented Architecture (SOA). In the e-government development process has 3 main stages, namely: database creation, service creation, and service implementation on the frontend. The results of this study are in the form of conclusions on the architectural capabilities used in e-government integration accompanied by the speed of services produced.**

Keywords*— *E-government, Integration, Service Oriented Architecture (SOA).

I. INTRODUCTION

The development of Information and Communication Technology (ICT) changes human life. This development changed the relationship between human relations, business, and changed relations between government and society. Rapid information needs make the government need to develop information technology. Like providing the best service for the community, internal government, partner government, and entity government. This service is commonly called E-Government.

According to the Ministry of Communication and Information, E-Government is an internet-based application of information technology and other devices managed by the government for the purpose of delivering information from the government to the public, business partners, and other institutions online [1]. E-Government Needs in Indonesia have become very important since the issuance of Presidential Instruction No. 3/2003 concerning National E-Government Development Policy and Strategy [2], which requires every regional region, institutions, government agencies, etc. to take advantage of the rapid advances in communication and information technology. For the sake of achieving efficiency, effectiveness, transparency and accountability in the administration of the government as well as the opening of opportunities for accessing, managing and utilizing information in large volumes quickly and accurately.

At present the development of E-Government in Indonesia is not yet maximal. Based on the 2016 survey, the United Nations (UN) published that Indonesia received a rating of 116 E-Government Development Index (EGDI) [3]. This is proven by 85% of the number of E-Government websites in Indonesia that can be accessed. While other websites are still inaccessible due to several factors, such as being repaired, server errors, failing to connect to the database, etc. And there are still many E-Government in Indonesia that have not implemented the Integration

System between services. Which results in a lack of effectiveness and efficiency of E-Government services.

Many regions in Indonesia have not implemented E-Government Integration, one of which is E-Government in Sidoarjo Regency. In Sidoarjo Regency in almost every service provides its own E-Government services. However, due to the absence of an integration system between E-Government services, the service process for both cross-official transactions and communication still uses manual methods so that it is less effective and efficient in terms of speed and accuracy.

Therefore E-Government in Sidoarjo Regency needs an integrated system to overcome these problems. With this system, it is expected that services in the government sector can run more effectively and efficiently. So that it can increase community satisfaction with government performance.

Based on the above background, the problems that can be formulated in this paper are as follows:

a. There is duplication of data on the E-Government server in Sidoarjo Regency which causes data swelling and depletion of storage space. Data duplication occurs because there are several tables with the same field between one application with another application.
b. The existence of data that is not synchronized between one service with another service due to user input errors. Because there is still a large amount of data that is not yet integrated, input data from users can result in differences when inputting to different applications that should have entered the same data.
c. Service in a government system that still requires a long time. This is because some government services must interact with several other agencies, and the interested community must go to the service.

II. ORIGINALITY

The purpose of this paper is that researchers will develop an integration system that is implemented in E-Government, Sidoarjo Regency using service oriented architecture (SOA). Where this integration system will connect various E-Government to different government agencies. So that later can accelerate and simplify the process of transaction and data communication as a form of service to the community. In addition, the aim of this paper is to utilize technological advances in the field of ICT with maximum management. With the aim of producing a service that is beneficial to the government and the people of Sidoarjo Regency.

III. RELATED WORK

As'ad [4] explained that integration between services is one of the most important problems affecting E-Government implementation throughout the world. Providing integrated services for citizens, businesses, and stakeholders involved in E-Government on a 'one stop portal' is considered a great opportunity for the government to improve the efficiency and effectiveness of their services. Successful E-Government is based on two factors. First is the availability of E-Government websites and services that are available not only during government working hours. The second is accessibility, which means that E-Government sites and services must be accessible to users everywhere. Dennis. et al. [5] explain to the government that there is a spread of data and information at each unit of the managing agency and the organizing agency of the activity. So with such cases there is data redundancy and difficulties in accessing data from one another. The uncoordinated data source is one of the problems that can be solved by using systems development that is integrated with e-Government. Data integration can only be carried out in certain physical scope, for example: the availability of a database management system, applications and databases connected to a computer network. Kayode, Ajibade Ibrahim [6]. Service Oriented Architecture (SOA) can radically change the order of application integration. Visibility, interaction, and effectiveness are the main concepts of implementing SOA. Besides that, SOA has gained high popularity in recent years because of its functions and services that make it possible to improve and expand existing software applications. In this paper it aims to integrate pharmaceutical information systems based on the concept of SOA with website services through a central service bus. The system from the database from the drug store will be integrated through the service bus so that the drug can be easily searched and the results will be displayed based on availability. From this paper the researcher concludes that SOA is very suitable for integrating software services in order to add effective and efficient software that is built. Hai, Henry. Et al [7]. The use of the Software as a Service (SaaS) application by the company is considered to be less than optimal.

Because companies require to buy and deploy infrastructure, paying for expensive resources for customization, upgrading and ongoing maintenance. Many find that SaaS applications require little or no infrastructure and maintenance, can be used quickly and have a cost model that can be predicted to have less risk and a faster return on investment. New demand has driven rapid innovation in SaaS applications, SaaS platforms, and SaaS integration tools. However, the company still has the burden of integrating this application with the back-office system and applications at their location. Complex corporate integration requirements even challenge the current best SaaS solution provider; there are still limitations and pitfalls to watch out for. In this paper the researchers describe some of the best practices of SaaS integration, present case studies, and highlight emerging integration technologies that can help ease the burden of integrated SaaS applications themselves. Sutanta, Edhy. Mustofa, Khabib [8]. In this paper, it is explained how the implementation of Service Oriented Architecture (SOA) in Bantul Regency. Besides also discussing in detail about web services and how the concept of E-government. The developed web service has three main components. First the provider that provides information and data. Both agents / brokers who build web services. Third is the requester that can use the services of the provider. One example of a case of web service needs in Bantul district government is synchronization and exchange of data between information systems. For example, MIS (Management Information System) Monograph online which requires population data. To meet the data requirements online SIM Monographs need to access the resident master data where these needs can be met by creating a web service. The design of web services between different systems is also explained in this paper in detail with clear images.

IV. System Design

The following in Figure 1 below is an ongoing paper system design. Starting from the process of creating a database, creating a web service and finally using the web service into the interface.

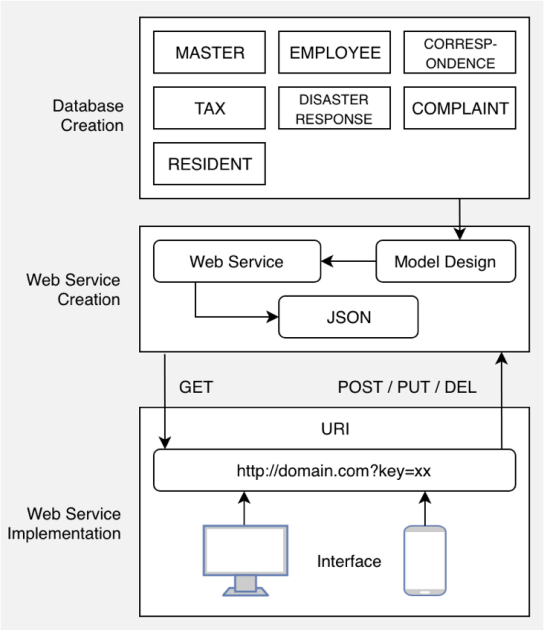

Figure 1. System Design

Making Web Services uses the Service Oriented Architecture (SOA). Namely by making service packages in the form of small units that can be developed continuously. This SOA will also be able to integrate between various other package services that have been made.

The first step to build this paper is to create a database. The researcher uses a MySQL database, because the database is a free-to-use database, and facilitates effective database management by connecting it to the database. By migrating current database applications to MySQL, many companies enjoy significant cost savings on new projects. Dependency and ease of management can save time solving problems that are wasted in fixing downtime problems and performance problems.

The second step is making a web service. At this stage, there are two stages, namely making a model design and making the web service itself. Next from the web service produces output in the form of JSON data (JavaScript Object Notation). The following is an explanation of these stages:

a) Model Design

Model design aims to represent a database in the form of code. The model built functions to connect the database with the web service for interaction in processing data.

The attributes contained in the model are almost the same as the attributes in the table, but are more specific. Figure 2 is an example of a code model that represents a table. In this paper

the researcher used the squelize library to simplify the modeling.

```
'use strict';
module.exports = (sequelize, DataTypes) => {
  const epbb = sequelize.define('epbb', {
    nik: {
      allowNull: false,
      type: DataTypes.STRING,
      validate: {
        notEmpty: true,
      }
    },
    total: {
      allowNull: false,
      type: DataTypes.STRING,
      validate: {
        notEmpty: true,
      }
    },
    jatuh_tempo: {
      allowNull: true,
      type: DataTypes.DATE,
      validate: {
        notEmpty: false,
      }
    },
```

Figure 2. Model Design in Code

From the example code above, it is initialized that the table name is *epbb*, from the *epbb* table containing the *nik* field, total and maturity. Each field has its own attributes. For example, *nik* has an allowNull attribute that is false, meaning that the *nik* field cannot be empty (must be filled). This *nik* field has a string data type or if the database is commonly called varchar. Then for validation it contains notEmpty value true, meaning that if the data entered in the *nik* field is empty then the model will automatically provide a feedback error on the service associated with the table.

For functions that support modeling, it usually depends on the library used. So it needs to be adjusted to the need for the application service creation process.

b) Web Service

After making a model the next step is to enter the core of making a web service. web service is usually called REST (REpresentational State Transfer), which is a web-based communication architecture standard that is often applied in developing web-based services. Generally using HTTP (Hypertext Transfer Protocol) as a protocol for data communication.

In the REST architecture, the REST server provides resources (data / resources) and the REST client accesses and displays these resources for future use. Each resource is identified by URIs (Universal Resource Identifiers) or global ID. The resource is represented in the form of text format, JSON or XML. Generally the format uses JSON and XML.

In this study there are several services built in the form of web services. The following is a service that will be built:

1) Correspondence
   Correspondence is an application used for making correspondence, such as the making of Individual Domicile Letters, Business Domicile Letters, Hospitality Permit, Village Certificate, Poor Certificate, Cover Letter, etc.
2) Population administration information system
   Population Administration Information System, to organize the system of population administration in Indonesia, this system includes population data collection and civil census. Population data include: Population Registration Number (NIK), Family Card (KK), Resident Identity Card (KTP), Birth Certificate, Death Certificate, etc.
3) Taxation
   Used to see a list of taxes and tax liabilities. The following is the result of checking a tax from someone by entering a Population Registration Number) which later can display the amount of tax this year and tax history in previous years.
4) Community Complaint Service Center
   services to accommodate the aspirations of the Sidoarjo people which in real time will be emailed to each member of the Regional Representative Council of Sidoarjo Regency.
5) Disaster Response
   Application of information systems about natural disasters and reports of natural disasters. In this waiter the public can send information related to the disaster that is happening through the website or mobile. This service is equipped with a google map platform to make it easier for people to determine the location of a disaster.
6) Employee Management Information System
   Service data center (data center) staffing in the Sidoarjo Regency Government that provides information about staffing.

Currently there are thousands of employee data recorded in the database.

In making a web service, must initialize how to process it. There are several methods for processing web services, namely, GET, POST, PUT / PATCH. GET functions to retrieve data, POST functions to add data and PUT to change data.

c) JSON (JavaScript Object Notation)

The output of a web service is in the form of JSON (JavaScript Object Notation) as a form of representation of the REST itself like Figure 3. JSON has a basic component, among others :

a) Objects: Objects begins and ends with curly braces ({}).
b) Object Member: Consists of strings and separated by commas (,).
c) Arrays: Arrays begin and end with square brackets ([]) and contain different values.
d) Values: Values can be string, object, array, or literal.
e) Strings: The string is surrounded by double quotes ("") and contains a colon Unicode (:). Members are separated by commas (,). Values separated by commas (,).

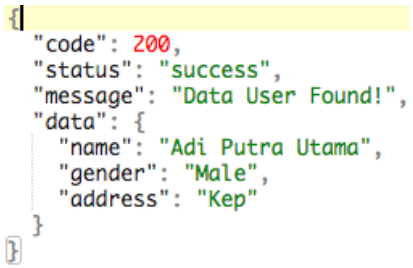

Figure 3. JSON Syntax

The final step is the implementation of web services to the user interface such as the web or mobile app. To implement a web service must go through the mechanism of using URI syntax (Uniform Resource Indentifier). The following is an explanation of the URI and the user interface that will be built.

a) URI (Uniform Resource Indentifier)

URI is a character that identifies web resources by name, location or both. In this case the resource in question is a web service. The URI has several components, namely: schema, autority (user, host, port, path), query, fragment information [9].

The following in Figure 4 is an example of URI syntax in the use of a web service that serves to retrieve population data based on unique number.

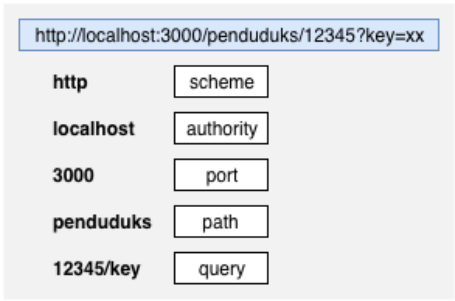

Figure 4. URI Syntax

Each URI starts with a scheme to assign identification to the character. There are many types of schemes that are very familiar, such as http, https, ftp, sftp, etc. Then for authority use localhost. This authority functions as the authority for naming URIs. Port section has a value of 3000 which is a subcomponent of authority. Usually by default the http scheme has port 80. For the path named inhabit according to the data to be retrieved through the syntax. While for parameters valued at 12345 which is the number of population data and added key as verification or permission to access the web service.

To facilitate the implementation of the web service or the use of a web service the researcher provides documentation that is complete and ready to use as an example in the Figure below which is a documentation of the use of the login function, namely by using the POST method accompanied by a body username and password.

b) User Interface

E-government will be developed in two interface platforms (user interfaces), namely on the web. This aims to achieve ease of use in e-government applications in Sidoarjo Regency.

V. Experiment and Analysis

In this experiment the researchers will measure the application process speed. To measure the speed of the application process, researchers make examples of applications that do not implement SOA and that implement SOA. In Figure 5 is an application without implementing SOA. The researcher made the application use the MySQL database and the Yii2 PHP framework.

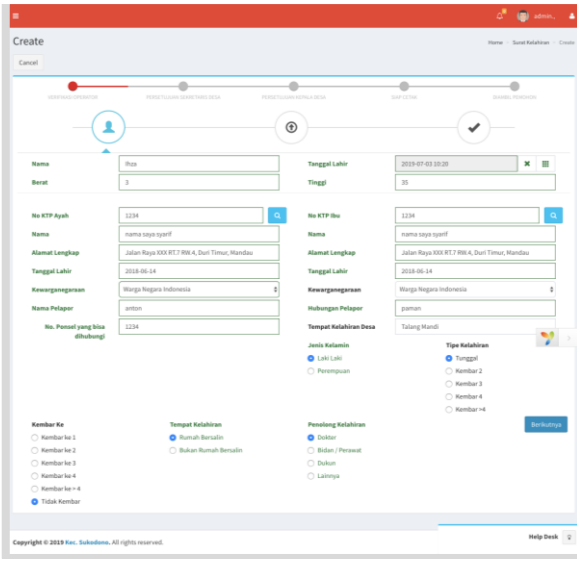

Figure 5. Application without SOA

Whereas in Figure 6 is an application that applies SOA. The researcher uses the MySQL database and framework express Node.JS.

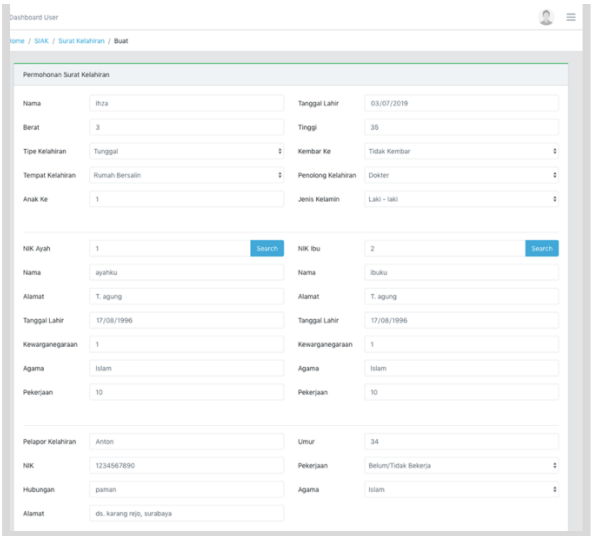

Figure 6. Application with SOA

From the two applications, researchers conducted a trial of input data on birth certificate services which had the same number of forms. From the experiment the researcher obtained results, namely the time needed for the application without applying SOA for data input, which is 0.24 seconds. While the time needed for applications that implement SOA for data input is 0.16 seconds. So it can be concluded that the speed of the application process is faster that applies SOA.

In addition, researchers also conduct speed trials on all services, the results of which can be seen in Table 1.

Services	Results of Experiments with SOA (in second)					Results of Experiments without SOA (in second)				
	1	2	3	4	5	1	2	3	4	5
Resident Card	0.02	0.02	0.025	0.02	0.02	0.032	0.034	0.35	0.033	0.033
Family Card	0.025	0.02	0.025	0.02	0.02	0.03	0.03	0.039	0.03	0.03
Birth Certificate	0.08	0.1	0.09	0.08	0.1	0.19	0.22	0.19	0.18	0.2
Death Certificate	0.033	0.035	0.04	0.036	0.036	0.046	0.047	0.048	0.043	0.044
Individual Domicile Letter	0.04	0.045	0.04	0.04	0.04	0.05	0.055	0.052	0.054	0.053
Individual Bussiness Letter	0.05	005	0.055	0.05	0.5	0.06	0.06	0.065	0.06	0.06
Crowd Permit	0.05	0.05	0.05	0.05	0.05	0.07	0.08	0.08	0.075	0.08
Village Clearance Letter	0.018	0.02	0.019	0.02	0.019	0.018	0.02	0.019	0.02	0.019
Certificate of Poor	0.02	0.022	0.024	0.02	0.02	0.035	0.032	0.034	0.038	0.037
Cover Latter	0.015	0.017	0.017	0.015	0.016	0.025	0.027	0.027	0.025	0.026
Taxation	0.013	0.014	0.013	0.013	0.014	0.023	0.024	0.023	0.023	0.024
Complaint	0.012	0.012	0.014	0.013	0.014	0.026	0.023	0.024	0.023	0.024
Disaster Response	0.014	0.015	0.015	0.015	0.014	0.024	0.027	0.025	0.025	0.026

Tabel 1. Speed Test Trial

From this it can be concluded that the SOA is faster than without the SOA. In terms of synchronization and lack of duplication of data, researchers make every service do not need to store data that already exists in other services. An example is the making of an resident card request. The applicant only needs to enter the number which can then be searched to the resident service to retrieve the data based on the nik. So that the data will not be duplicated and the data will always be synchronous.

In developing this governance, the researcher made how the application that was created did not cause data duplication. Data duplication is closely related to the database schema created. This means that it is very concerned about each field in the table created. As the example in Figure 7 is an example of overcoming data duplication. Complaint services and services for certificates are different services, but have the same data requirements, which both require population data. To get population data, complaint services only need to access GET web service residents. Likewise with the provision of village certificates only need to access the GET web service residents. On the other hand, applications that are built have synchronous data, because the data taken comes from the same place or web service.

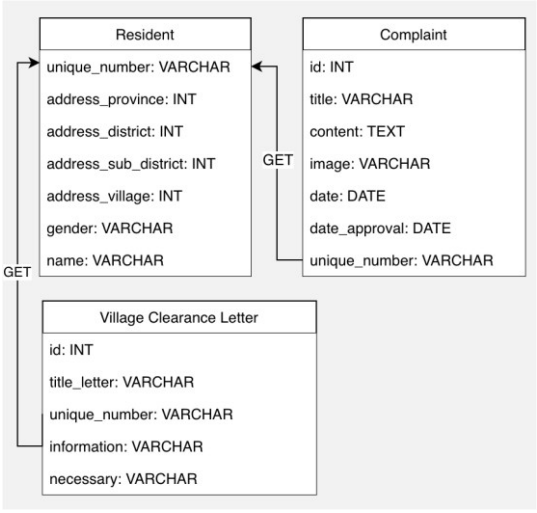

Figure 7. Database Schemes for Overcoming Data Duplication

VI. Conclusion

E-government is one service that can accelerate the process of transactions and communication between government officials and the community. However, due to the lack of data integration the service process of e-government has become less than optimal. Because some services also depend on other services. In addition, because there is no integration, it is possible that data can experience data duplication, which also allows for data that is out of sync.

From this research e-government applications have been integrated with the SOA. The e-government manufacturing process has three main stages. First is the database creation process. The database must be made with due regard to the correct scheme so as not to cause duplicate data. Second is the process of making web services. Making web services also pay attention to what tools are used, such as the programming language used, the web server used, etc. Third is the implementation of web services into the user interface, both in the form of a web application or mobile application.

With the development of integrated e-government applications using the SOA, the process can be run faster, because this architecture makes the application into the form of a web service. So that it can connect between one service and another. In addition, there is duplication of data and the presence of asynchronous data can be handled properly.

Acknowledgment

The researcher thanked the local government of Sidoarjo Regency. And thanks to the party who helped me complete this paper.

References

1. Putri, Tania Dwika, Penerapan Electronic Goverment (E-Gov) Melalui Penyelenggaraan Situs Web Pemerintah Provinsi Riau Untuk Mewujudkan Transparansi Publik. Pekanbaru: Universitas Riau. 2015.
2. Instruksi Presiden Republik Indonesia Nomor 3 Tahun 2003 Tentang Kebijakan dan Strategi Nasional Pengembangan E-Government.
3. Widowati, Dyah P. D, Inilah Peringkat E-Government Indonesia Berdasarkan Survei PBB 2016. https://bpptik.kominfo.go.id/2016/09/09/2190/inilah-peringkat-e-governmen t-indonesia-berdasarkan-survei-pbb-2016/. Accessed 15 May, 2018
4. As'ad, et al. Importance of service integration in e-government implementations. Sheffield Hallam University Research Archive. 2016.
5. Dennis, et al. Integrasi Sistem Informasi Pada Pemerintahan Dengan E-Government. Jakarta. Universitas Bina Nusantara. 2014.
6. Kayode, Ajibade Ibrahim. Using SOA with Web Services for effective data integration of Enterprise Pharmaceutical Information Systems. UK/Ghana. University College. 2013.
7. Hai, Henry. Et al. 2009. *SaaS and Integration Best Practices.* FUJITSU Sci. Tech. J.
8. Sutanta, Edhy. Mustofa, Khabib, Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov Di Pemkab Bantul Yogyakarta. Yogyakarta. JURTIK - STMIK BANDUNG. 2012.
9. T. Berners-Lee, et al. Uniform Resource Identifier (URI): Generic Syntax. Network Working Group. 2005.