Service-Oriented Architecture for E-Marketplace Model Based on Multi-Platform Distributed System

Yaddarabullah1*, M F Muttaqin2, M Rafiansyah3
1,2,3 Faculty of Creative Industry and Telematics, Universitas Trilogi, Indonesia

Email: *yaddarabullah@trilogi.ac.id

Abstract. E-Commerce has become a solution in online transactions. Based on data from the Central Bureau of Statistics of Indonesia, in 2016, there were 26.2 million users and continued to grow by 17% from 2006 to 2016. On the other hand, the development of e-marketplace has also grown significantly in Indonesia. There are 100 trillion transaction value in 2018. Currently, e-commerce actors are still unable to compete with existing e-marketplace. This is due to limited stock and types of products sold by e-commerce. One of the challenges faced by e-commerce managers is increasing the number of customers and transactions. The purpose of this study produced an architecture for data integration between e-commerce to the e-marketplace. The multi-platform e-marketplace model was developed that was integrated and could distribute products from several e-commerce. Service oriented architecture was used as a method in integrating and distributing products from e-commerce to e-marketplace. Based on the results of testing with 90 request can be concluded that the process of data distribution from e-commerce to e-marketplace can be done in a fast time with average loaded time of 0.72 seconds and the size of data accessed is relatively small with packed average size of 5.68 KB. The conclusion of this study remains the use of service-oriented architecture can be applied for e-marketplace that integrated with several e-commerce for supporting some feature like online payment and courier.

1. Introduction

Nowadays, e-marketplace has grown rapidly. This is indicated by the data released by the Central Bureau of Statistics of Indonesia. In 2016, there were 26.2 million transactions and continued to grow by 17% from 2006 to 2016. E-marketplace is used as a platform that serves to meet the sellers with the buyers to do online transaction [1]. The e-marketplace agents do not provide the products they sell, but provide access to conventional sellers to have accounts and trade their products in e-marketplace. Conventional sellers who have joined the e-marketplace can easily arrange the storefronts of the products sold. Research by Naibaho [2] and Suriyanto [3], shows the advantages of the existing e-marketplace include providing multi-platform web-based and mobile applications to make it easier for buyers to make transactions. Besides, the e-marketplace has a function that is integrated in non-cash payments with the online payment providers and integrated with online courier providers in delivering the products. In the previous research by Ciarniene [4], it shows that e-commerce actors are still unable to compete with existing e-marketplace. This is due to limited stocks and types of products sold by e-commerce and there are some e-commerce that limit services in certain areas, making it difficult to compete with e-marketplace [5][6][7]. Research by Jauhari [8] and Amin e [9], show another disadvantage of e-commerce is that not many have been integrated into the online payment system and the shipping system through online courier. Distributed system software was a software that connects two applications in the form of client side and server side [10]. Related research about distribute and
integrate data [11][12], show that service-oriented architecture can be used for integrate and distribute data, research by Zhu [13], which applies service-oriented architecture for integrate and distribute data from buyer to third party that handle payment, another research by Ferguson [14], which used IBM service-oriented architecture to increase functional interfaces between components and products visible through a service model. Based on the problem and previous research, the purpose of this study to analyze and produced an architecture of e-marketplace in which there is a function to integrate products from various e-commerce and distribute data with online payment and online courier services. Service-oriented architecture will be used as method for distribute data from e-commerces to e-marketplace and support for multi-platform services.

2. Method
The stages of this research are started from functional requirements analysis, data communication architecture design, program code preparation, and testing.

2.1. Functional Requirements Analysis
The first stage in this study was the functional requirements analysis of e-marketplace. The main requirements in e-marketplace include presenting product catalogs, purchase, payment and product shipping transactions. The results of the analysis were used in designing data communication architectures in the system.

2.2. Data Communication Architecture Design
The second stage was designing data communication architecture (request, process, response). Designing data communication architecture was based on service-oriented architecture. There are three main functions in distributing data including e-commerce to e-marketplace, from e-marketplace to online payment and from e-marketplace to online courier. The data were accessed between platforms using the Application Program Interface (API). The request from e-marketplace to e-commerce API, online payment API and online courier API was done via HTTPS protocol. The data provided by e-commerce API, online payment API and online courier API to e-marketplace were in XML format. System architecture that was distributed in multi-platform from e-marketplace is illustrated in the figure 1.

The e-commerce API provides six services that can be accessed by e-marketplace in presenting product data. The e-marketplace dashboard will display product data provided by e-commerce. This

![Figure 1. Service-Oriented Architecture-Based Data Communication Architecture](image)
will save database server capacity on e-marketplace. The function used to access product data is a Service List of Product. The following is how to access the function.

Address: 
https://renjershop.000webhostapp.com/server/serviceAllJersey.php

Output:
<data_allJersey>
    <nama kode_barang="20" harga="200000" g1="brg-1544789733.png" rating="5"> 
        Jersey AC Milan
    </nama>
    <nama kode_barang="35" harga="350000" g1="brg-1544789898.png" rating=""> 
        Jersey Real Madrid
    </nama>
</data_allJersey>

The second e-commerce API that will be accessed by e-marketplace, which is Service Price, functions to get the price of a product in the following way.

Address: 
https://renjershop.000webhostapp.com/server/serviceJerseyHarga.php?tampungg1=170000&tampungg2=219999

Output:
<data_jerseyHarga>
    <nama kode_barang="20" harga="200000" g1="brg-1544789733.png" rating="5"> 
        Jersey AC Milan
    </nama>
</data_jerseyHarga>

The third e-commerce API that is accessed is a service Brand that functions to select products based on brands. The following is how to access the function.

Address: 
https://renjershop.000webhostapp.com/server/serviceJerseyMerk.php?tampungg=Adidas

Output:
<data_jerseyMerk>
    <nama kode_barang="20" harga="200000" g1="brg-1544789733.png" rating="5"> 
        Jersey AC Milan
    </nama>
</data_jerseyMerk>

The fourth e-commerce API that is accessed is the Product Code service that functions to get the product descriptions based on the product code.

Address: 
https://renjershop.000webhostapp.com/server/serviceJerseyProduct.php?idp=20

Output:
<data_jerseryKodeProduct>
    <nama hargap="200000" desSingkat="" desPanjang="" addInfo="" 
        g1="brg-1544789733.png" g2="" ratings="5" merk="adidas"tahun=""> 
        Jersey AC Milan
    </nama>
</data_jerseryKodeProduct>
The fifth e-commerce API accessed, the Product Ratings service, functions to get a list of products based on the product ratings. There is one parameter used, which is the rating value.

Address:
https://renjershop.000webhostapp.com/server/serviceJerseyRating.php?tampungg=5

Output:
<data_jerseyRating>
  <nama kode_barang="20" harga="200000" g1="brg-1544789733.png" rating="5">Jersey AC Milan</nama>
</data_jerseyRating>

The sixth e-commerce API that is accessed, the Product Search service, functions to get the product list based on the product search.

Address:
http://renjershop.000webhostapp.com/server/serviceJerseySearch.php?tampungg=jersey

Output:
<data_jerseySearch>
  <nama kode_barang="20" harga="200000" g1="brg-1544789733.png" rating="5">Jersey AC Milan</nama>
  <nama kode_barang="35" harga="350000" g1="brg-1544789898.png" rating="">Jersey Real Madrid</nama>
</data_jerseySearch>

The online payment API provides two services for processing transactions from e-marketplace. The first function is Service Payment Check which functions to receive and process all payments made. There is one parameter that is sent to find out the payment, namely the payment code. The following is how to access and the output generated from the Payment Check API.

Address:
http://alfiaansaputra.000webhostapp.com/server/serviceCekPembayaran.php?kodee=8872636255

Output:
<data_cekPembayaran>
  <kodee tgl_transfer="16/02/2019">8872636255</kodee>
</data_cekPembayaran>

The second function, namely the Balance service, is to receive and process all balance top-up data. The users of e-marketplace can use the balance service that has been backed up to their account. The following is the access and the output generated from the Balance service.

Address:
http://alfiaansaputra.000webhostapp.com/server/serviceCekTopUp.php?kodee=9928374628111

Output:
<data_cekPembayaran>
  <id_topup kode_transfer_saldo="9928374628111">1</id_topup>
</data_cekPembayaran>
Online courier API provides two services to process shipments of the products from e-marketplace. The first function is that the Shipping Price service aims to receive and process shipping service fee data. The following is the access and output of the Shipping Price service.

Address:
https://rizalcourier.000webhost.com/server/serviceKurir.php

Output:
<data_kurir>
  <paket harga="9000">Reguler (2-3 hari)</paket>
  <paket harga="15000">Today (1 hari)</paket>
</data_kurir>

The second function, namely the Shipping Status service, aims to determine the status of the products sent based on the invoice. The following is the access and output of the Shipping Status service.

Address:
https://rizalcourier.000webhostapp.com/server/serviceTerimaAll.php

Output:
<data_kurirCheck>
  <invoice status="Sudah Diterima">50012</invoice>
  <invoice status="Sudah Diterima">50014</invoice>
  <invoice status="Sudah Diterima">50016</invoice>
  <invoice status="Sudah Diterima">50018</invoice>
</data_kurirCheck>

2.3. Program Code Preparation
In this stage, the program code was prepared using the PHP programming language and using the MySQL Server database. The data were distributed using XML format.

2.4. Testing
The testing was performed by using the blackbox method to find out whether every function in the e-marketplace, namely the process of accessing product data, purchase and shipping transactions could run well. Then, testing was done to the performance of e-marketplace and the distribution of data using GT Metrix.

3. Results and Discussion
The results of this study were formed by an e-marketplace that can be integrated with other platforms. Service-oriented architecture was implemented at some e-commerce. Function testing was done to find out whether the main functions in e-marketplace could work well. Testing was done by connecting three e-commerce to the e-marketplace (see Table 1).
**Table 1. Functional Testing**

| Test Code | Test Unit | Information | Result |
|-----------|-----------|-------------|--------|
| U1        | Displaying E-commerce Product Catalogs | This function is used to display all products owned by e-commerce of Renjershop, Merchands Store and Dekoryuk through the API provided by each e-commerce. | OK |
| U2        | Selecting products based on price ranges | This selection function is used to filter e-commerce products based on price ranges | OK |
| U3        | Selecting products based on brands | This selection function is used to filter e-commerce products based on product brands. | OK |
| U4        | Displaying products through the selected product code | This function is used to display product details based on the product code selected. | OK |
| U5        | Selecting products based on the rating value | This selection function is used to filter e-commerce products based on product ratings. | OK |
| U6        | Product search | The search function is used to search for products that contain names in the search column. | OK |
| U7        | Checking and processing all payments | This function is used to process all transactions made through e-marketplace websites. | OK |
| U8        | Checking all packages of shipping services provided | This function is used to find out the shipping service package done by online courier. | OK |
| U9        | Checking product shipping status | This function is used to know the status of product shipments | OK |

Table 1 explain about the results of functional testing. All functional of services that provide by e-marketplace which had been tested work well. First unit testing display products from e-commerces. E-marketplace save e-commerces API URL and request list of the products then e-commerce return data of the product using XML format. Function for filtering products was handle by unit test code U2, U3, U5 and U6. When filtering for data of the products, the e-commerce API that requested from e-marketplace would include parameters such as the price ranges, brands, code, rate and name. The next test is carried out at the time of the transaction. The product purchased will be put into the shopping cart and will get the invoice number and transfer code. This transfer code is used for payments of the transactions. The payments are made through online payment services provided by the online payment platform. The customers must have an account in the online payment platform first, then they can make the transaction payment by submitting the transfer code that has been obtained from e-marketplace. The products shipping will be integrated with online courier. Checking the status of the shipping can be done through the invoice number obtained from the e-marketplace. The next was the testing of the performance of the system that had been developed. Performance testing was done by using GT Metrix, which was accessed through http://gtmetrix.com. The tested performance included the percentage of page or service speed accessed, processing time, and page size. The testing was performed to the ten available services, namely on the e-commerce API, online payment API and online courier API. There were four variables that were tested on the ten services, including the **Page Speed Score**, **YSlow Score**, **Loaded Time** and **Page Size**. The testing was done by making requests as much as 90 times in every two hours for each service. The test of **Page Speed**
Score consist of several testing for speed measurement, while e-marketplace access the data from e-commerce. First to know the following requests were returning 404/410 responses. Either fix the broken links or remove the references to the non-existent resources. Second to minify HTML for the following resources to reduce the size of data by 3B. The test of YSlow Score for a grades web page based on one of three predefined rulesets or a user-defined ruleset. Loaded Time obtained from the sum of DNS lookup, connecting, sending, waiting and receiving time. The following is the graph of loaded time testing.

![Figure 2](image.png)

**Figure 2.** Graph Testing All Service of E-Marketplace Architecture

Based on the observations of the service testing that had been done, it could be seen that the Loaded Time average of the six services provided by the e-commerce API List of Product (S1), Price (S2), Brand (S3), Product Code (S4), Product Ratings (S5), Product Search (S6) was 0.73 seconds for fully loaded and 0.6 for onload time. The Loaded Time average of the online payment API (S7, S8) was 0.65 seconds for fully loaded and 0.5 for onload time. The Loaded Time average of the online courier API (S9, S10) was 0.86 seconds for fully loaded and 0.7 for onload time. This figure shows that the time needed to handle the request to the service was relatively fast. The next stage was drawing conclusions from the results of testing of the ten services stated in the following table.

**Table 2. Performance Testing**

| No | Service           | Page Speed Score | YSlow Score | Loaded Time | Page Size |
|----|-------------------|------------------|-------------|-------------|-----------|
| 1  | List of Products  | 99%              | 99%         | 0.8s        | 5.69 KB   |
| 2  | Price             | 99%              | 99%         | 0.6s        | 5.67 KB   |
| 3  | Brand             | 99%              | 99%         | 0.7s        | 5.67 KB   |
| 4  | Product Code      | 99%              | 99%         | 0.6s        | 5.70 KB   |
| 5  | Product Ratings   | 99%              | 99%         | 0.7s        | 5.67 KB   |
| 6  | Product Search    | 99%              | 99%         | 0.9s        | 5.69 KB   |
| 7  | Payment Check     | 99%              | 99%         | 0.7s        | 5.64 KB   |
| 8  | Balance           | 99%              | 99%         | 0.5s        | 5.64 KB   |
| 9  | Shipping Price    | 99%              | 99%         | 1.1s        | 5.64 KB   |
| 10 | Shipping Status   | 99%              | 99%         | 0.6s        | 5.65 KB   |
Table 2 explain about the results of tests on the performance of API services on e-commerce. It show that with the number of accesses 99 times in the same time to e-commerce, it will get a relatively fast response. This is indicated by the average value of page speed score is 99% with an average loaded time value of 0.71 seconds and the average size of data obtained is 5.68 KB. Based on the results of these trials, the use of service-oriented architecture concepts in the multi-platform e-marketplace model can distribute products on many e-commerce sites into one e-marketplace. The process of distributing product data from e-commerce to e-marketplace is relatively fast with small file sizes. Each transaction on e-marketplace can be distributed to online payment services by checking transaction invoices. The process of checking payments can also be done from e-marketplace with fast processing time and small data size. Transactions in e-marketplace can also be traced to the goods delivery provider by checking the transaction invoice.

4. Conclusion
Based on the results of system development and testing done, it can be concluded that the use of service-oriented architecture can be used as a way to distribute multiplatform data. By having an e-marketplace model that can distribute data from e-commerce equipped with online payment and shipping services, the e-marketplace model can be used as a centralized media in trading products of e-commerce. Further research that needs to be done is to integrate various e-commerce and then do a comparative analysis of transactions conducted in the e-commerce and in the e-marketplace. This is done in order to know the effectiveness of this e-marketplace model.

References
[1] W.Srisadono 2018 Strategi Perusahaan E-Commerce Membangun Brand. Pustaka Komunikasi, 1(1) pp. 167-179
[2] B. R. Betti Dane Hutauruk, JF Naibaho 2017. Analisis dan Perancangan Aplikasi Marketplace Cideramata Khas Berbasis Android. Jurnal Method, 3(1) pp.242-246
[3] A. D. Suriyanto 2017. Komunikasi Tanda Badge (Emblem) antara Konsumen dan Pemilik Toko Online Pada Marketplace. Komunikasi, 3(1) pp.49-55
[4] Ciarniene and Stankeviciute 2015. Theoretical Framework of E-Business Competitiveness. International Scientific Conference Economics and Management, pp. 734 – 739
[5] Wandoko. 2016. Online trust building through third party trust transfer and third party protection. International Conference on Computing and Applied Informatics pp. 1-8
[6] Katawetawaraks and Wang. 2011. Online Shopper Behavior: Influences of Online Shopping Decision. Asian Journal of Business Research, 1(2) pp.66-74
[7] Singh 2010. Global E-Commerce: A Portal Bridging The World Markets. Journal of Electronic Commerce Research, 11(1) pp. 1-5
[8] Jauhari. 2011. Aplikasi M-Commerce Untuk Penjualan Buku Pada Suatu Toko Buku Berbasis WAP (Wireless Application Protocol) Service Dengan Media Handphone. Jurnal Sistem Informasi, 3(2) pp.346-358
[9] B. Z. Amin and P. Amin 2012. Ebay is the World’s Online Marketplace. IOSR Journal of Business and Management, 2(7) pp.23-29
[10] Maikibaeva. 2016. Development of Distributed System for Electronic Business Based on Java-Technologies. International Journal Of Environmental & Science Education, 11(10) pp. 3861-3883
[11] T. Sanwal, S. Avasthi, and S. Saxena. 2016. E-Commerce and its sway on the minds of young generation. International Journal of Scientific and Research Publications, 6(3) pp.112-117
[12] D. L. Yaddarabullah. 2018. Perancangan Sistem Komunikasi Data Alat Pencatatan Meter. Jurnal Infotekjar, 3(1) pp.49-54
[13] S.H. Zhu and T.Kuang. 2014. E-commerce model and platform based on the third-party. Journal of Chemical and Pharmaceutical Research, 6(3) pp. 863-868
[14] Ferguson. 2005. Service-oriented architecture: Programming model and product architecture. IBM Systems Journal, 44(4) pp.753-780