

# Aksyaa- Android Application Accounting Information System

Bayu Hermawan, Giri Wahyu Wiriasto\*, Lalu A. Syamsul Irfan Akbar

Dept Electrical Engineering, University of Mataram  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: bayuhermawan957@gmail.com, [giriwahyuwiriasto, irfan]@unram.ac.id

\*Corresponding Author

**Abstract** - Today, software technology in accounting is rapidly advancing due to the increasing use of mobile devices and the rise in new businesses. Previously, we proposed an Accounting Information System (AIS) that operates on a web platform as the primary server. The mobile application's development also utilizes the same database as the web version, functioning on both web and mobile platforms. The Application Programming Interface (API) for the AIS system was developed using the RESTful method, serving as an integration medium via the HTTP protocol for web and mobile communication in a case study involving company transaction data. The API implementation is explained in detail regarding the mobile interface development. In the HTTP communication cycle, the request-response mechanism plays a central role. Clients request data from the server, and the server responds by sending the requested data. Integration testing confirms that the mobile application functions correctly and displays information accurately. Similarly, testing involved inputting 798 transaction records and 141 account lists into the system. The mobile page successfully showcases all features, presenting data output consistent with the web-based AIS. These features encompass reports on purchases and sales, cash flow, general ledger, trial balance, and income statement.

**Key words:** Accounting Information System; RESTful API; Android Application

## I. INTRODUCTION

Advances in information technology are something that cannot be avoided because technological advances are in line with the development of science. Information technology can be applied in all areas of life [1][2]. Many digital-based financial services exist today, one of them is mobile payment services [3]. One form of current technological advances is a web service which is a system that works in an integrated manner using supporting technologies such as SOAP, REST, WSDL or WDDX [4] and using the HTTP, SMTP or FTP protocols [5]. For Internet applications, the RESTful interface paradigm is now widely used, as it provides a consistent, scalable, and flexible model for a wide variety of interfaces [6]. Web services are moving towards new technologies leading to migration of SOAP to RESTful API [7]. The term "web service" cannot be separated from API, because API is a technology used to facilitate the exchange of information between two or more applications. While a web service is

a technology used to facilitate the exchange of information between two or more applications over a network/network.

The Application Programming Interface (API) was created to design a simple, flexible and efficient interface for programmatically requesting and interacting with databases based on REpresentational State Transfer (REST) principles [8]. REST APIs allow users to directly access data via Hypertext Transfer Protocol (HTTP), and provide users with an efficient way to interact with databases and other web services [9]. RESTful Web services reduce resource consumption and are more efficient for deploying and provisioning Web services from resource-constrained mobile devices [10].

Currently there is an Accounting Information System (AIS) running on a website platform that can process accounting data and present various kinds of financial information [11][12]. The financial information presented in the AIS is in the form of reports on purchases and sales of goods, cash flow reports, general ledger reports, trial balance reports, and profit and loss reports. The accounting information system website has provided a web service that will be used for data integration with other systems. The web service uses REST API supporting technology that exchanges data using the HTTP protocol. The web service architecture using REST API is to exchange data centered on the Request-Response cycle. Request is the process of requesting data by the client to the server, while the response is data sent by the server according to the client's request. The provided REST API has implemented several API methods, namely GET, PUSH, PUT, and DELETE [13]. The method used can be accessed through the Uniform Resource Identifier (URI) [14] to be able to make resource requests to the server. The resources provided in the form of resources in this AIS system case study include 'accounts', 'goods' resources, 'purchasing' resources, 'sales' resources, 'general journal' resources, 'general ledger' resources, 'trial balance' resources, and 'profit and loss' resources. After the mobile apps platform makes a resource request, the server will provide a response in the form of data in JSON format. The JSON format is highly recommended as a REST API response format, as it is more readable and CPU friendly compared to the XML format [15].

The REST API is provided by representing the database contained on the server. Making a REST API, the first thing to do is to prepare the data needed such as sales data, purchases, and other data. Next determines the endpoint format used for interaction between the API and the server using the HTTP protocol through the Uniform Resource Identifier (URI). The final stage is to determine the method based on the endpoint to be used. The available methods are GET (read resource), POST (create resource), PUT (update resource), DELETE (delete resource). Below is an example of the endpoints and methods used.



Fig. 1. Endpoints and sales methods

From the description and brief description of the accounting information system and REST API that has provided, this study intends to create a mobile-based accounting information system application that will run on an Android device. This application will integrate data using the REST API that has provided so that the data in the application will be the same as that in the web version.

## II. RESEARCH METHOD

### A. REST API Architecture

Fig. 2 shows the REST API architecture which describes the design of the REST API used in application development.

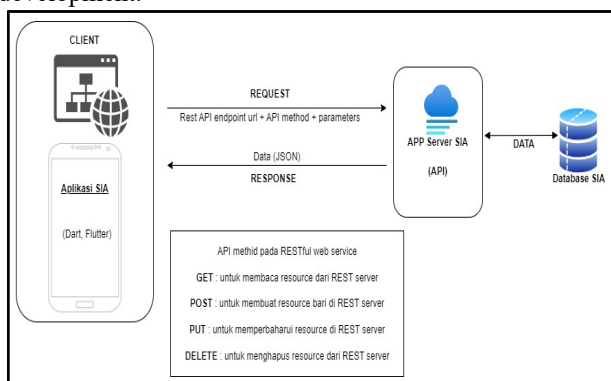


Fig. 2. REST API Architecture

Fig. 2. shows the REST API architecture which consists of client, server, and database. The application to be created will implement a RESTful API, where the Client (website and mobile) sends HTTP requests to the server via the URI along with the methods and parameters used. The methods that can be used are GET, POST, PUT, DELETE. After the client makes a request, the server will provide response data in the form of JavaScript Object Notation (JSON) format. Data received from the API via the Post method is written to the database. For data

requests using the Get method, the API reads from the database [26].

### B. Physical Data Model (PDM)

This stage is the database design stage where all the data in the system will be stored in one database, so that the data becomes integrated. PDM is already a physical form of database design that is ready to be implemented into the Database Management System (DBMS) [16][17] of an accounting information system.

Having a dedicated framework lifecycle specifically designed for mobile applications is crucial due to the complexity involved in their development. This framework, known as the Mobile Application Development Lifecycle (MADLC), plays a vital role. The MADLC consists of several phases, namely Identification, Design, Development, Prototyping, Testing, and Maintenance. This comprehensive lifecycle has been successfully implemented for over a year, specifically in the development of Android mobile applications [27].

Fig 3. shows the PDM of the aksyaa.com system which consists of 25 related tables which are also the aksyaa mobile database. The main tables include the 'general journal', 'purchasing journal', 'sales journal', and 'ledger' tables.

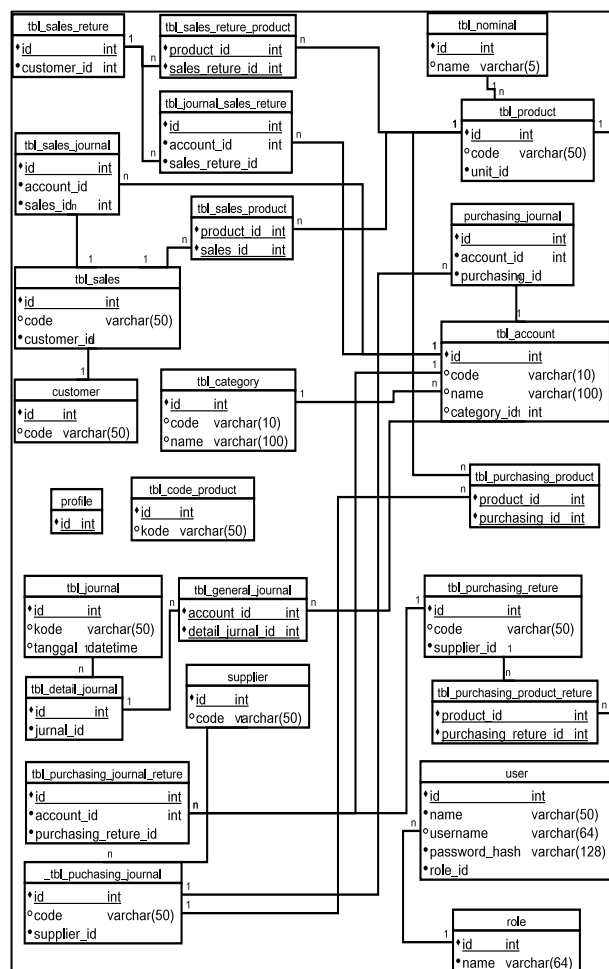


Fig. 3. Physical Data Model of aksyaa.com [25]

C. Use Case and Class diagram

UML design diagram of AIS of aksyaa.com. Fig. 4 shows the design use case diagram depicting user interaction with the internal system of the aksyaa.com system before implementing sharia principles.

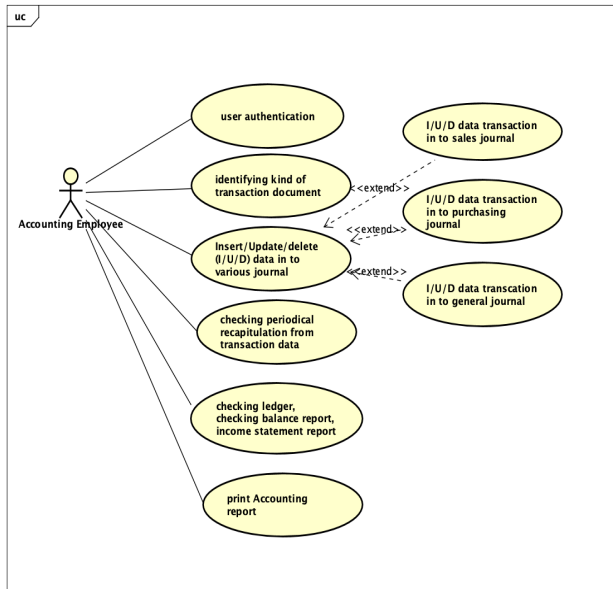


Fig. 4. Use case diagram of aksyaa.com - Accounting Information System (AIS) [25]

Next, class diagrams are part of the recommended UML to describe the class of the system to be created [18]. Class diagrams show the structure of the software and the logical relationships between classes statically [19][20]. Fig. 5 shows the class diagram used to describe the software system in this study. There are 18 classes, some of which already have relationships. These classes will be used as the basic schema to make it easier when creating program code.

As shown in Fig. 5, there are several main classes, namely the 'general journal' class, the 'purchasing journal' class, the 'sales journal' class, and the 'general ledger' class and other classes.

D. User interface of AIS web version

The user interface is an important element of the information system from the user's perspective [21]. The accounting information system on the website has provided a simple interface and can make it easier for users to interact directly with the system. The SIA website has several user interface pages including the 'account list', 'purchase', 'sales', 'ledger', 'balance sheet' and 'income statement' pages.

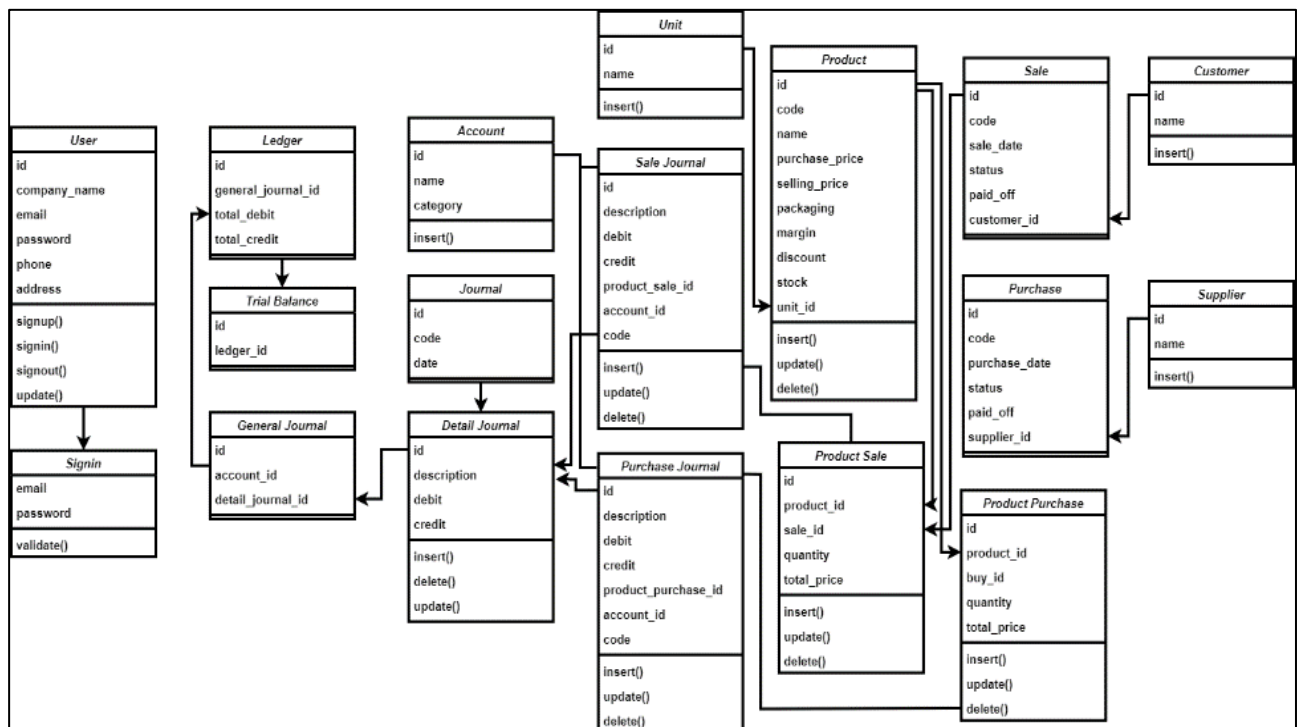


Fig 5. Class diagram of aksyaa mobile version

Kode Akun	Nama Akun	Debit	Kredit
1-10001	Kas	418.162.482	
1-10002	Rekening Bank	387.472.200	
1-10200	Persediaan Barang	934.108.339	
1-10705	Aset Tetap - Perengkapan Kantor	77.000	
3-30000	Modal Saham		10.703.000
3-30001	Tambahan Modal Disetor		1.916.720.000
4-40000	Pendapatan Jasa		716.146.000
4-40100	Diskon Perjualan	50	
4-40101	Gaji	22.280.000	

Fig. 6. Web side interface for trial 'balance sheet'

Kode Akun	Nama Akun	Debit	Kredit
1-10001	Kas	418.162.482	
1-10002	Rekening Bank	387.472.200	
1-10200	Persediaan Barang	934.108.339	
1-10705	Aset Tetap - Perengkapan Kantor	77.000	
3-30000	Modal Saham		10.703.000
3-30001	Tambahan Modal Disetor		1.916.720.000
4-40000	Pendapatan Jasa		716.146.000
4-40100	Diskon Perjualan	50	
4-40101	Gaji	22.280.000	

Figure 7. List of 'Account'

Fig. 6 shows the user interface of the AIS web side for the 'balance sheet' page. It contains information on all transaction records based on the account type. In this page there is also provided a feature to filter transaction data by account or transaction time and download 'balance sheet' reports based on filter results.

### III. RESULT AND DISCUSSION

The results and discussion will discuss the results of the application design that has made which includes a discussion of how the system works the REST API method and the results of testing of several supporting components.

#### A. Testing result with various account of data transaction

The transaction data test is a test of transaction data based on accounts on the AIS web version. There are 141 accounts provided and for each account there are derivative accounts from the main account, including 'assets', 'liabilities', 'capital', 'income' and 'expense'.

Fig. 7 shows a list of accounts that have used or filled in based on samples of transaction data such as 'cash' accounts, 'bank accounts', 'service income', 'goods inventory', 'capital' and other accounts. The nominal in these accounts is automatically filled in after the transaction data is entered. As an example of a sales transaction that can be seen in Fig. 8. The transaction data contains 4 product items in the 'sales' account, including 'Springbed', 'Foam Mattress', 'Panel & General', and 'Sofa' products with a total transaction quantity of 798 records. This data is inputted into the AIS system. As shown in Fig. 6, the 'service income' account accumulates according to the total 'sales' shown in Fig. 7 which is IDR 716,146,000.

NO NOTA	TANGGAL	NAMA customer	NAMA BARANG	Q	T	F	HARGA SATUAN	TOTAL
965	30-Oct-20	SIGMA RSG AC	PANEL & UMUM	1			200,000	200,000
966	30-Oct-20	WISATA SY SRI	PANEL & UMUM	1			1,300,000	1,300,000
967	30-Oct-20	RINI	MAT SUPER DREAM 100 (ST)	1			3,700,000	3,700,000
968	30-Oct-20	CASH	MILD TR SB	1			350,000	350,000
969	30-Oct-20	CASH	KURSI DPMO	10			240,000	2,400,000
970	30-Oct-20	MIRNAMA	MARI MASTER 800 HCPT	1			2,050,000	2,050,000
971	30-Oct-20	CASH	AA 36 SW	5			2,000,000	10,000,000
972	30-Oct-20	BUDISMAN	MAT CENTRAL GRAND DLX 120	1			1,000,000	1,000,000
973	30-Oct-20	MELU	ALCAMA RE	1			475,000	475,000
974	30-Oct-20	MARWAN	MATRAT CENTRAL DLX 120	1			1,150,000	1,150,000
975	30-Oct-20	CASH	BL BECCA	1			375,000	375,000
976	30-Oct-20	AMROZI	MATRAT CENTRAL DLX 120	7			600,000	4,200,000
977	30-Oct-20	CASH	BL CLASH MOTIF	7			50,000	350,000
978	30-Oct-20	CASH	BRAND BRICE	1			500,000	500,000
979	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
980	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
981	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
982	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
983	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
984	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
985	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
986	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
987	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
988	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
989	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
990	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
991	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
992	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
993	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
994	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
995	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
996	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
997	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
998	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
999	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
1000	30-Oct-20	CASH	BL CLASH MOTIF	1			50,000	50,000
GRAND TOTAL				798				716,492,000

Fig. 8. Sample of raw transaction data for testing

#### B. Create of RESTFULL API script as a result

In this stage, the mobile version of the AIS application program code is written using the Dart programming language with the Flutter framework. Flutter is a user interface structure created by google to create interfaces on android and IOS [22][23]. Some of the features that are made at this stage are 'add product', 'product update', 'sales transaction', 'general journal transaction', display 'general ledger', 'trial balance', and 'income statement'. Below are some snippets of program code using the flutter framework.

#### C. Application integration with the API as a result

Application integration with the RESTFULL API serves to connect two or more different platforms, so that data from the two platforms are connected to each other. The integration process starts from determining the API endpoint and the method to be used to access the API. Then the endpoint is inserted into the function in the program code so that it can be connected to the API. The following below are some of the results of application integration with the API displayed in the program code. Some of the program code displayed is the API integration program code for 'trial balance', 'ledger', 'profit and loss', and 'sales'.

```

var url = Uri.parse('${baseUrl}neraca?tanggal=$date&filter_by=$period');
var headers = {'Content-Type': 'application/json', 'X-API-KEY': apiKey, };
try {
    final response = await http.get(url, headers: headers);
    if (response.statusCode >= 200 && response.statusCode < 300) {
        List data = jsonDecode(response.body);
        List<TrialBalanceModel> trialBalances = [];
        for (var item in data) {
            trialBalances.add(TrialBalanceModel.fromJson(item));
        }
        return trialBalances;
    } else if (response.statusCode >= 500 && response.statusCode <= 599) {
        // print(response.statusCode);
        throw response.statusCode;
    } else {throw response.statusCode;}
} catch (e) {rethrow;}
    
```

Fig. 9. API's script code for 'balance sheet'

• API's for mobile apps integration for 'balance sheet'

Fig. 9 and Fig. 10. shows a samples snippet of the API script for the integration process on a 'balance sheet', 'ledger'. The following is a brief explanation, in the script there is an endpoint URI from the 'balance sheet' class which is used to request data to the server. It can be seen that the methods and parameters used are using the GET method to retrieve resources from the 'balance sheet' provided from the API on the server side. If the 'balance sheet' resource request exists, the server will send a response in the form of 'balance sheet' data in JSON format. The data will be stored in a 'balance sheet' class model which will then be displayed on the 'balance sheet' page.

• API's for mobile apps integration for 'ledger'

```

var url = Uri.parse('${baseUrl}buku-besar');
var headers = {'Content-Type': 'application/json', 'X-API-KEY': apiKey, };
try {
    final response = await http.get(url, headers: headers);
    if (response.statusCode >= 200 && response.statusCode < 300) {
        List data = jsonDecode(response.body);
        List<LedgerModel> ledgers = [];
        for (var item in data) {
            ledgers.add(LedgerModel.fromJson(item));
        }
        return ledgers;
    } else if (response.statusCode >= 500 && response.statusCode <= 599) {
        throw response.statusCode;
    } else {throw response.statusCode;}
} catch (e) {rethrow;}
    
```

Figure 10. API's script code for 'ledger'

A. Result of UI aksyaa mobile version

You can see the user interface (UI) resulting from the development of the mobile version of the AIS system. Designing the form of the User interface is made using the Figma application [24], then creating the program code using the Flutter framework. The results of the application design realization that have been made are splash screen pages, dashboard pages, products and product details, general journals, ledgers, balance sheets, and income

statements. The following successive images of the final UI form of the AIS system can be seen in Fig. 11 to Fig. 16.

Fig. 11. shows the dashboard page which contains information related to the number of sales per year, monthly sales charts, sales summaries, and balances of several accounts contained in the balance sheet. Furthermore, in Fig. 12 shows the general journal page which contains data on the results of daily transactions separated by transaction accounts. On this general journal page there are also two features to filter data by date and year, and there is a button to add new general journal data.

Furthermore, in Fig. 13 shows the general ledger page which contains a collection of accounts that are used to summarize transactions that have been recorded in the general journal. The data displayed is based on the results of the account filter, where the list of data from the account contains transaction, debit and credit dates. On this ledger page there are also two features to filter data by month and year, and there is a button to download ledger reports based on account filter results.

In Fig. 14 shows the balance sheet page which contains all types of account names along with the total balance of each account arranged sequentially based on account code. On this balance sheet page there is also a feature to filter daily, monthly and yearly data, and there is a button to download a balance sheet report based on the results of the data filter. Then Fig. 15. shows the income statement page which contains financial reports in monthly and annual periods containing income and expenses. The figure shows that the service revenue in the case study data obtained is IDR 708 million, where this value is obtained from total sales during the annual period. The value of profit divided by gross loss is obtained from the subtraction between revenue and cost of sales of IDR 708 million. The value of operational expenses is obtained from expense accounts such as salaries, wages, overtime and others amounting to IDR 13 million. From these values, a profit can be generated divided by a net loss from the reduction between income and expenses.

Finally in Fig. 16 shows the sales page which contains a list of sales transaction data. On this sales page there is a feature to do based on status, namely success and failure. In general, blockbox testing has been utilized in the testing of the Aksyaa mobile system.



Fig. 11. Dashboard page



Fig. 12. General journal page



Fig. 13. Ledger page

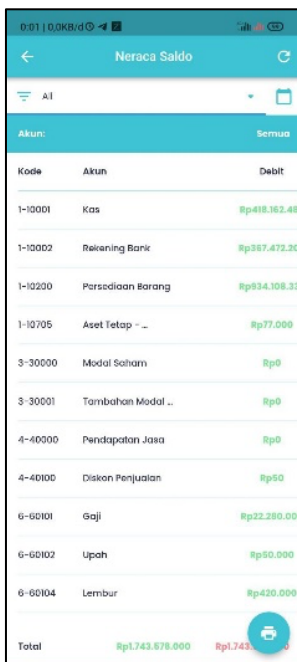


Fig. 14. Balance sheet page



Fig. 15. Incoming statement page



Fig. 16. Sales page

#### IV. CONCLUSION

Based on the results of system testing that has made and used blackbox-testing, the conclusion is that the mobile version of the Aksyaa AIS application can integrate data with the web version of the system using the REST API method. A snippet of the API script using the Flutter framework has also provided by implementing the design from the class diagram as an example of real integration in developing mobile-based applications. The class diagram design is used to describe software systems and as a basic

scheme in making program code. System testing was carried out by entering 798 case study transaction data records into the system with 141 account lists provided. As a result, after integrating the mobile page view, it shows that all features are able to work and display the same output data as the web version of AIS, including features for reports on purchases and sales of goods, cash flow, general ledger, trial balance, and income statement. Based on the data displayed on these pages, a financial information is obtained that can be used as material for decision making as an information executive.

## ACKNOWLEDGMENT

This publication support by LPPM University of Mataram.

## REFERENCES

- [1] Wójcik, W., & Sikora, J. (2017). Recent advances in information technology. In *Recent Advances in Information Technology*. <https://doi.org/10.1201/9781351243179>
- [2] Kamanga, R., & Alexandra, P. M. (2019). Facilitated Adoption of Accounting Information Systems: A First Step to Digital Transformation in Township Microenterprises. 2019 Open Innovations Conference, OI 2019. <https://doi.org/10.1109/OI.2019.8908236>
- [3] Prihono, O. F., & Sari, P. K. (2019). Comparison Analysis Of Social Influence Marketing For Mobile Payment Using Support Vector Machine. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*. <https://doi.org/10.22219/kinetik.v4i4.921>
- [4] Tihomirovs, J., & Grabis, J. (2017). Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics. *Information Technology and Management Science*, 19(1). <https://doi.org/10.1515/itms-2016-0017>
- [5] Ma, Z., Reynolds, J., Dickinson, J., Wang, K., Judd, T., Barnes, J. D., Mason, J., & Bailey, M. (2019). The impact of secure transport protocols on phishing efficacy. 12th USENIX Workshop on Cyber Security Experimentation and Test, CSET 2019, Co-Located with USENIX Security 2019.
- [6] Prehofer, C., & Gerostathopoulos, I. (2017). Modeling RESTful Web of Things Services: Concepts and Tools. In *Managing the Web of Things: Linking the Real World to the Web*. <https://doi.org/10.1016/B978-0-12-809764-9.00004-4>
- [7] Padmanaban, R., Thirumaran, M., Anitha, P., & Moshika, A. (2022). Computability evaluation of RESTful API using Primitive Recursive Function. *Journal of King Saud University - Computer and Information Sciences*, 34(2), 457–467. <https://doi.org/10.1016/j.jksuci.2018.11.014>
- [8] Ong, S. P., Cholia, S., Jain, A., Brafman, M., Gunter, D., Ceder, G., & Persson, K. A. (2015). The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. *Computational Materials Science*, 97. <https://doi.org/10.1016/j.commatsci.2014.10.037>
- [9] Arcuri, A. (2019). RESTful API automated test case generation with Evomaster. *ACM Transactions on Software Engineering and Methodology*, 28(1). <https://doi.org/10.1145/3293455>
- [10] Mohamed, K. E., & Wijsekera, D. (2012). Performance analysis of web services on mobile devices. *Procedia Computer Science*, 10, 744–751. <https://doi.org/10.1016/j.procs.2012.06.095>
- [11] Dalci, I., & Tanis, N. V. (2013). Benefits Of Computerized Accounting Information Systems On The JIT Production Systems. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 13(1).
- [12] Neogy, T. K. (2014). Evaluation of Efficiency of Accounting Information Systems: A Study on Mobile Telecommunication Companies in Bangladesh. *Global Disclosure of Economics and Business*, 3(1). <https://doi.org/10.18034/gdeb.v3i1.170>
- [13] Singh, A., & Jeyanthi, N. (2020). MVP Architecture Model with Single Endpoint Access for Displaying COVID 19 Patients Information Dynamically. *Proceedings - 2020 12th International Conference on Computational Intelligence and Communication Networks, CICN 2020*. <https://doi.org/10.1109/CICN49253.2020.9242573>
- [14] Shieh, J., & Reese, T. (2015). The Importance of Identifiers in the New Web Environment and Using the Uniform Resource Identifier (URI) in Subfield Zero (\$0): A Small Step That Is Actually a Big Step. *Journal of Library Metadata*, 15(3–4). <https://doi.org/10.1080/19386389.2015.1099981>
- [15] Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML data interchange formats: A case study. 22nd International Conference on Computer Applications in Industry and Engineering 2009, CAINE 2009.
- [16] Nyerges, T. (2017). Physical Data Models. *Geographic Information Science & Technology Body of Knowledge*, Q1. <https://doi.org/10.22224/gistbok/2017.1.1>
- [17] de Jong, K., & Karssenbergh, D. (2019). A physical data model for spatio-temporal objects. *Environmental Modelling and Software*, 122. <https://doi.org/10.1016/j.envsoft.2019.104553>
- [18] Berardi, D., Calvanese, D., & De Giacomo, G. (2005). Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2). <https://doi.org/10.1016/j.artint.2005.05.003>
- [19] Fauzan, R., Siahaan, D., Rochimah, S., & Triandini, E. (2021). Automated Class Diagram Assessment using Semantic and Structural Similarities. *International Journal of Intelligent Engineering and Systems*, 14(2). <https://doi.org/10.22266/ijies2021.0430.06>
- [20] Gonçalves, L. J., & Farias, K. (2019). Towards a hybrid approach to measure similarity between UML models. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3330204.3330226>
- [21] Ilyas, Q. M., Ahmad, M., Zaman, N., Alshamari, M. A., & Ahmed, I. (2022). Localized Text-Free User Interfaces. *IEEE Access*, 10. <https://doi.org/10.1109/ACCESS.2021.3139525>
- [22] Idan Arb, G., & Al-Majdi, K. (2020). A Freights Status Management System Based on Dart and Flutter Programming Language. *Journal of Physics: Conference Series*, 1530(1). <https://doi.org/10.1088/1742-6596/1530/1/012020>
- [23] Ernawati, S., & Wati, R. (2021). ANDROID-BASED QURAN APPLICATION ON THE FLUTTER FRAMEWORK BY USING THE FOUNTAIN MODEL. *Jurnal Riset Informatika*, 3(2). <https://doi.org/10.34288/jri.v3i2.205>
- [24] Rully Pramudita, Rita Wahyuni Arifin, Ari Nurul Alfian, Nadya Safitri, & Shilka Dina Anwariya. (2021). Penggunaan Aplikasi Figma Dalam Membangun Ui/Ux Yang Interaktif Pada Program Studi Teknik Informatika Stmik Tasikmalaya. *Jurnal Buana Pengabdian*, 3(1), 149–154.

- <https://doi.org/10.36805/jurnalbuanapengabdian.v3i1.1542>
- [25] Giri Wahyu Wiriasto, Misbahuddin, Muhamad Syamsu Iqbal, Djul Fikry Budiman, Sjamsjiar Rachman, Lalu Ahmad Syamsul Irfan Akbar. "Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science) (MIMSE-I-C-2022, Atlantis Press, 367-378, 2352-538X, 2022, [https://doi.org/10.2991/978-94-6463-084-8\\_31](https://doi.org/10.2991/978-94-6463-084-8_31).
- [26] G. W. Wiriasto, R. W. S. Aji and D. F. Budiman, "Design and Development of Attendance System Application Using Android-Based Flutter," 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), Surabaya, Indonesia, 2020, pp. 1-6, <https://doi.org/10.1109/ICVEE50212.2020.9243190>
- [27] Vithani, Tejas and Kumar, Anand."Modeling the mobile application development lifecycle". Proceedings of the international Multiconference of Engineers and Computer Scientis 2014 Vol1, IMECS 2014, March'12 - 14, Hongkong. ISBN:978-988-19252-5-1, ISSN:2078-0966. [https://www.iaeng.org/publication/IMECS2014/IMECS2014\\_pp596-600.pdf](https://www.iaeng.org/publication/IMECS2014/IMECS2014_pp596-600.pdf)