AnalevR: An Interactive R-Based Analysis Platform as a Service for Utilizing Official Statistics Data in Indonesia

Keywords: R, analysis environment, platform, collaboration tools, services.

1. INTRODUCTION

The process of making decisions, drawing conclusions, and estimating outcomes require fast and easy access to up-to-date and reliable information. As Indonesia's national statistical agency, BPS-Statistics Indonesia produces a massive amount of wide-range strategic data every year. The use of these official statistics data has expanded to non-government groups such as researchers, students, and businesses. However, these data are still underutilized by the public due to technical limitations (lack of skilled and experienced employees) and raw data exclusivity and locality (distributed and separately-stored microdata). Other issues such as bureaucratic procedure, long waiting time, and the prices to buy the data have also contributed to worsening the situation.

In line with Indonesia's National Bureaucratic Reform Program, BPS-Statistics Indonesia aims to create innovations in public services by modernizing the way people benefit from the data. We introduce AnalevR, an online R-based analysis platform for accessing, analyzing, and visualizing official statistics data for free without having to own the original microdata. Data and analysis modules are put in a cloud storage and can be explored via the menu provided. The analysis is performed inside the workspace, either using Graphical User Interface (GUI) mode (menu and dialog) or non-GUI mode (syntax editor). AnalevR executes the R-based codes remotely and displays the result in the output container. We use R as the underlying engine because it has a complete collection of libraries for analysis and visualization compared to other languages. All user-defined variables and functions are automatically saved in the workspace for future use.

2. METHODS

2.1. Identifying User Needs

We conducted a small survey in June - July 2018 to identify what people expect from an analysis tool and how they want to benefit from BPS-Statistics Indonesia data.

2.2. System Architecture

AnalevR architecture (Figure 1) consists of three components: 1) Server for executing the codes, 2) Client for facilitating interaction between user and system, and 3) Message Broker for interconnecting the Server with the Client.

2.2.1. AnalevR Server

The Server comprises Session Worker, Session Manager, and Output Transformer. A Session Worker handles an R session and interacts with the database and file system to read the data and file needed in processing the request. Each session worker has a workspace that stores all information related to that particular session, including 1) Session file (.Rdata) that contains saved variables and command history; 2) Temporary files such as an image of a plot that will be displayed on a browser or a csv file that will be rendered as a table.

AnalevR aims to handle multiple users at once. Since R is a single-threaded programming language, we need a Session Manager to manage numerous session workers and route the data traffic from and to the corresponding session worker. Furthermore, we also need output transformer to converts the AnalevR's output into a browser-displayable format such as converting R dataframe to CSV or transforming R plot to Base64 PNG.

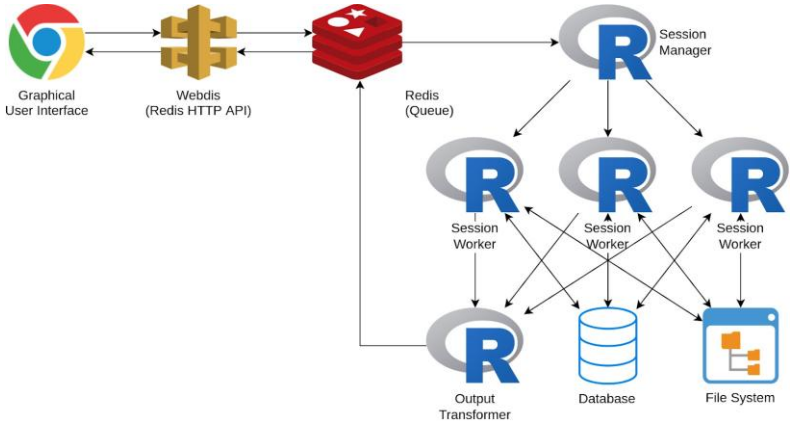

Figure 1. AnalevR System Architecture

2.2.2. AnalevR Client

AnalevR Client is a web-based interface designed with PHP and ReactJS [1] and implements the object-oriented paradigm to make the codes reusable and easy to maintain. It is also independent of the Server and only contains scripts for creating the user interface. This condition allows developers to construct different kinds of an interface according to the type of user devices, such as Android or IOS devices.

2.2.3. Message Broker

Message Broker sends messages using *Push-Pull* pattern [2] that allows a real-time and more reactive communication between Server and Client. We choose Redis [3] as a Message Broker for its fast and lightweight characteristics and Webdis [4] as a proxy to extend Redis capability in accessing Web Application Programming Interface (API).

2.3. AnalevR Module

AnalevR is preloaded with several analysis modules where each module consists of: 1) a javascript file to design GUI, and 2) R file(s) to handle the logic part. To more accommodate user needs, we facilitate users who understand R and Javascript to contribute to creating new analysis modules which can be utilized by others for further collaboration through the developer page.

3. RESULTS

In this section, we present the preliminary test results of both Server and Client using the data from National Socio-Economic Survey [5] and Labor Force Survey [6]. These data have been preprocessed through adjustments and cleaning. The number of data provided will continue to grow to ensure the variety of datasets that meet public requirements.

3.1. Testing on The Server Side

Two experiments are conducted to test Server's capability in handling requests. The first test is *Execution Time Test* to measure how long it takes for the Server to process requests on average. We run ten different scripts in sequence and each of them is repeated 100 times. We record the time spent for each script at each repetition and calculate the average and standard deviation value. The second test is *Concurrency Test* to examine the Server performance (speed and reliability) when dealing with multiple requests at once. The experiment is carried out three times with different numbers of concurrences: 10, 50, and 100 concurrent requests.

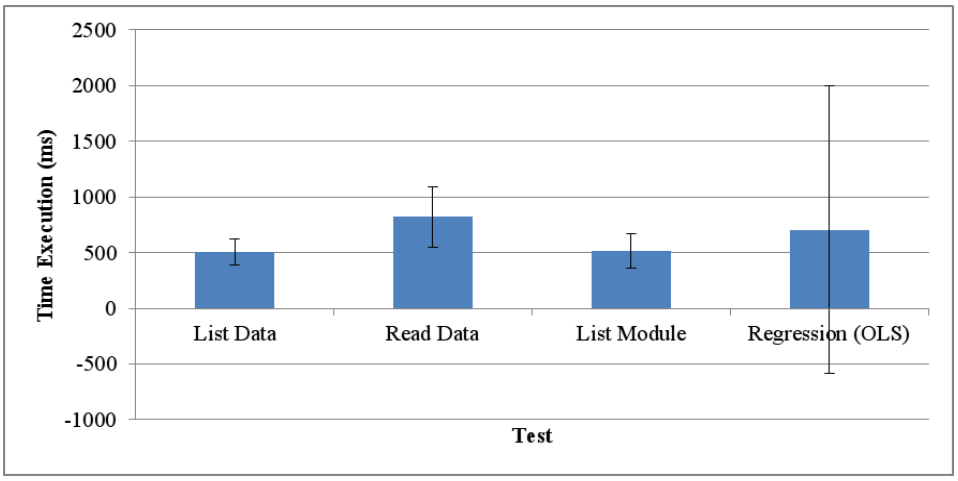

Figure 2 Result of Execution Time Test

The first test results show that AnalevR has a fast execution time. It can be seen from the average execution time for procedures list data, read data, list module, and regression analysis of 0.51, 0.82, 0.52, and 0.70 seconds, respectively as depicted in Figure 2. In terms of variance, the test shows the similar result of 0.12, 0.27, and 0.15 seconds for procedures list data, read data, and list module, respectively. The variance for regression analysis is slightly higher of 1.29 seconds. But this is still considered acceptable as analysis is a more complex task compared to other three procedures.

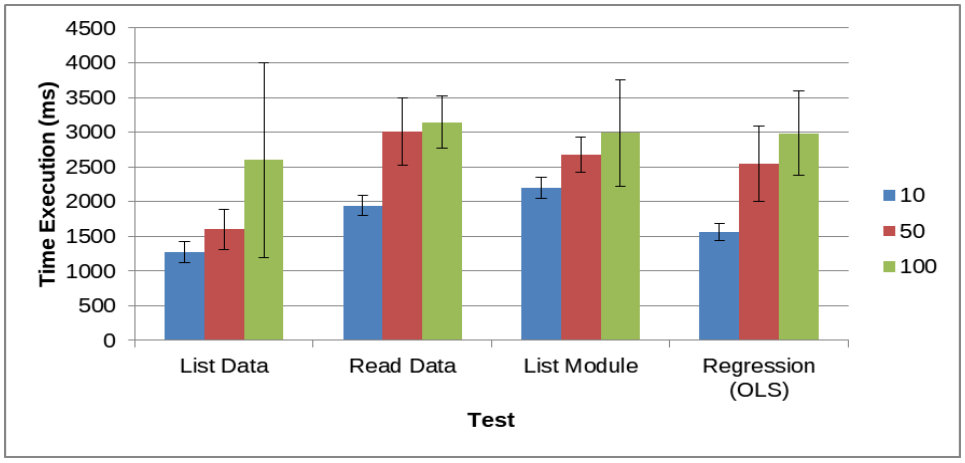

Figure 3 Result

The second test result (Figure 3) shows that AnalevR is reliable to handle multiple tasks at once. However, the more requests it manages, the more time it needs to complete all the requests. The test on conducting 100 concurrent requests of regression analysis shows that AnalevR can fulfill each request in 2.98 seconds on average.

3.2. Testing on The Client Side

A User Acceptance Testing (UAT) is conducted by asking 30 users to try AnalevR and fill a questionnaire about their understanding and satisfaction towards the platform. The users are given a brief description of what AnalevR is and how to use it. According to the survey, all respondents say that in general, they agree that AnalevR is easy to use and can help improving data utilization. However, it still needs some improvement in the number of analysis modules provided.

4. CONCLUSIONS

We present a breakthrough that can increase data utilization rate in BPS. AnalevR comes with many benefits: 1) All-in-one concept - support acquiring, wrangling, analyzing, and visualizing data, 2) Increasing efficiency by reducing the time lag between data request and analysis, 3) Flexibility - free access, easy to operate, multiple workspace, support both R users and non-R users, 4) Sustainability - support user collaboration.

By using a variety of technologies, AnalevR has the ability to provide better service and performance compared to similar tools that only use single technology. However, from the system administrator viewpoint, it causes the implementation becomes more complicated since every component must be installed separately. AnalevR is currently at an experimental stage and will continue to be developed and refined to better meet user needs. The prototype is up and running on http://simpeg.bps.go.id/analev-r and ready to use for analysis and visualization to gain insight from the data. Furthermore, it also creates the opportunity for professional communities to get involved in promoting Statistics and data utilization by contributing to building the modules. This project is open source with code available on https://github.com/erikaris/analev-r.

We believe that AnalevR will also be of interest to other countries that suffer from the same data underutilization problems regarding the bureaucracy and regulation issue to obtain the data. For future work, we plan to make it compatible with other languages such as Python and Java. This innovation will raise user involvement in employing BPS' data, promote the use of R, and ultimately increase statistical quality in Indonesia.

REFERENCES

[1] Facebook Inc. React - A JavaScript library for building user interfaces. 2018. Accessed April 10, 2018. Retrieved from https://reactjs.org/.

[2] Wong, K., Wang, C. 1999. Push-Pull Messaging: A High-Performance Communication Mechanism for Commodity SMP Clusters. *Proceedings of the 1999 International Conference on Parallel Processing*, pp.12

[3] Redis Labs. Redis. 2018. Accessed April 29, 2018. Retrieved from https://redis.io/.

[4] Nicolas Favre-Félix. Webdis - A fast HTTP interface for Redis. 2011. Accessed April 30, 2018. Retrieved from http://webd.is/.

[5] BPS-Statistics Indonesia. Survei Sosial Ekonomi Nasional 2017 Maret (KOR). Accessed August 25, 2018. Retrieved from https://microdata.bps.go.id/mikrodata/index.php/catalog/814.

[6] BPS-Statistics Indonesia. Indonesia - Survei Angkatan Kerja Nasional 2017 Februari. Accessed August 25, 2018. Retrieved from https://microdata.bps.go.id/mikrodata/index.php/catalog/802.