Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 22, No. 1, November 2022, pp. 229∼240
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v22i1.1325

Comparison of Memory Usage between REST API in Javascript and Golang

Hafizd Ardiansyah[1], Agung Fatwanto[2]
[1,2]Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia

Article Info

Article history:

Received July 07, 2022
Revised September 30, 2022
Accepted October , 2022

Keywords:

Comparison
Golang
Javascript
Rest API
Memory Usage

ABSTRACT

Various mobile devices have limited memory, thus it must be used as effectively as possible. As a result, apps that will operate on mobile devices must take memory usage efficiency into account. The REST API, which is typically used to connect several applications that utilize different types of technology so that the applications can be connected, is one sort of technology that is currently commonly used to construct mobile applications. Javascript and Golang are the types of technology used to create the REST API. Undoubtedly, each of these technologies offers a unique performance. Research that can give a broad overview of the variations in the impact of memory resource utilization between Javascript and Golang is therefore required. In this work, two REST APIs are created using Javascript and Golang by researchers utilizing an experimental quantitative methodology. Following that, the memory utilization of the two REST APIs was evaluated using the exact same two types of datasets obtained from console.cloud.google.com. There was a difference in memory consumption between Javascript and Golang after the Wilcoxon test, t-test for paired data, and equivalence test, but the difference was essentially inconsequential (practically insignificant).

Corresponding Author:

Agung Fatwanto,
Program Studi Magister Informatika,
Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia
Email: agung.fatwanto@uin-suka.ac.id

1. INTRODUCTION

The development of information technology continues to grow every day and the need for the internet continues to grow. Almost all companies need the internet to connect with information systems in the form of web applications or mobile applications. In this day and age, web applications and mobile applications are connected to each other so that they can exchange information and connect to each other via the internet [1].

Its ability to transmit data or information across platforms is due to its utilization of an environment distinct from other apps. The hypertext transfer protocol (HTTP) protocol is used by multi-platform applications because it facilitates multi-platform data interchange, such as web apps that can be accessed from all current devices [2]. Web application developers can use the API (Application Programming Interface) to connect the two applications using HTTP methods like get, post, put, and delete. HTTP allows application developers to connect web applications with mobile applications so that data exchange between the two applications can be done quickly and precisely [3]. A programming language that enables web application development, such as JavaScript, Golang, and other programming languages that support web application development, is required to aid in the multi-platform flow of data through web apps [3]. Javascript is a programming language that may be used on both the client and server sides of an application. Node.js is required for Javascript to execute on the server side [4], nonetheless, Golang is a server-side programming language [5].

The client side or client side is where JavaScript is utilized as a scripting language. Javascript is a language that is used on websites to create online apps that are more enticing. Initially designed only to run on the client side, node.js enabled the development of Javascript to operate on the server side as well [6]. Node.js made its debut in 2009; it is an event-driven, non-blocking programming language that allows online application developers to leverage HTTP to create user-accessible websites and applications [7]. Golang is an open-source programming language that boosts productivity in application development. Go is a simpler and more effective programming language overall [5]. Go itself provides features for garbage collection, concurrent operation, structure, and others [8]. There are several apps that can be created using Golang, one of which is a web application that can use HTTP to be accessible by people with the devices they use [9]. For the purpose of exchanging data between web apps and mobile applications, developers can use a server-side programming language to take advantage of the HTTP protocol by building a REST API (Representational State Transfer Application Programming Interface) website application [10].

The creation of REST APIs has been the subject of numerous studies. Golang and the New Simple Queue are implemented in research by A. Kristanto, et al. on a sandbox system based on REST API [11]. H. did research A performance comparison of the three employed programming languagesPHP, Python-web, and Node.jswas done by Brar et al [12]. Additional research have contrasted the effectiveness of RESTful utilizing Java with spring boot and C# and.net core [13]. The performance of the Web Backend and Database utilizing the Javascript programming language, Golang, and using the MySQL database, as well as MongoDB, was compared in research by F. Effendy, T. Taufik, and B. Andhilaksono [14]. According to one of the surveys, conducted by stackoverflow in 2020, Javascript received a percentage of 18.5%, whereas Golang received a percentage of 17.9%.

Now, commonly used applications communicate with one another via REST APIs and the HTTP protocol. One of them is a well-known e-commerce application that makes use of the REST API to build online and mobile applications that may communicate with one another [7]. On the devices utilized, such as laptops, desktop computers, and smartphones, each running application uses memory. Operating systems and running programs use memory. Each device has a certain amount of memory, so memory usage is a crucial factor that must be handled as effectively as possible [15].

The purpose of this study is to compare how memory is used by REST APIs created with Javascript and Golang technology. Prior to the development of the Golang programming language, comparisons in earlier studies had only been made using a small number of long-established programming languages. The programming language Golang has not been compared in many studies. To compare how much memory the REST APIs created with the Javascript and Golang programming languages utilize, see this study's comparison of memory usage.

Using a browser with the inspect capability, we will compare how much memory is used by the Javascript and Golang REST APIs in this study. The author uses a dataset from Google Cloud, which offers the different sorts of datasets required, to support this research. The GET protocol is the sole one employed by the HTTP protocol in this study.

2. RESEARCH METHOD

2.1. Research Type

This study uses experimental quantitative methodology. The primary data used in the analysis was gathered by the researcher independently conducting a number of tests. The equivalency test, T test for paired data, and Wilcoxon test were then used to statistically examine the data gathered from the experimental outcomes.

2.2. Research Flow

Javascript and Golang are two separate technologies that are used in the construction of REST APIs. Experimental quantitative research methods are used in this process, and the results show how well Javascript and Golang perform when it comes to utilising memory resources. The HTTP GET protocol will be used to retrieve the data [7]. Figure 1 depicts the steps of the research that will be done for this study.

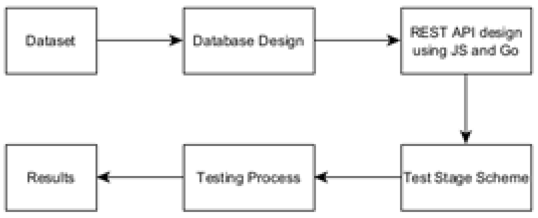

Figure 1. Research Flow

The research stages that were created will later help with the creation of REST APIs and testing to get findings from comparisons of the memory resource efficiency of Javascript and Golang technologies.

2.3. Research Dataset

Finding a dataset for comparison study on the functionality of the REST API, which was created using two distinct technologiesJavascript and Golangis the goal of this stage. The dataset utilized is a public dataset that is meant for research and is dispersed around the internet. According to this study, the dataset is only available on console.cloud.google.com. Large amounts of various sorts of data are made available on the internet, and the dataset will eventually be transformed to JSON format [16][17]. Table 1 contains the address for the dataset provider.

Table 1. URL Dataset

Dataset	
Google Cloud	https://console.cloud.google.com/bigquery?project=meta-altar-292605&supportedpurview=project&j=bq:US:bquxjob_5c5c024e_1752abe3236&page=queryresults

The data type used by each of the datasets, namely the string data type or in the form of text data, is the same for both the USA Person data and the Wikipedia data. The difference between the two datasets is the length of the data; for the USA Person data, the length of the data in table 2 is not greater than 10 characters, while for the Wikipedia data, the length of the data is greater. Table 2 shows the data specifications from the two data sources and the total number of data points used in this investigation, which was 100,000 for the USA Person and Wikipedia data.

Table 2. Dataset Specification

Dataset	Specifications	
USA Person	Name	State
Wikipedia	Label	Description

2.4. Database Design

At this point, a MySQL database will be established with the names of each table's fields and tables, which are organized into two tables and two fields, respectively, for each dataset used. Data that will be used later will be kept in the database. In Figure 2, you can see more information.

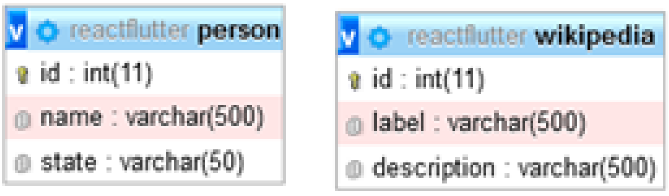

Figure 2. Database Design

2.5. REST API Design

Currently, Javascript and Golang are the two programming languages being used in the REST API design process. Exclusively the GET method, which only gets database data with a response status code of 200 OK, is employed by the HTTP method in this study [18].

The REST API will be used in two different scenarios, with the first scenario having the REST API run locally and the second scenario having it run online. The amount of data used in all scenarios is the same: 100,000 data for each dataset.

2.6. Test Scheme

The test plan is designed to establish a flow for the testing process that will subsequently be applied to the REST API, which was created by combining Javascript and Golang, two programming languages. The following are the plans or steps used during testing:

A Development process and stages

(a) Research creates REST API utilizing Javascript and Golang, two programming languages that connect to pre-existing data in the database; the data used comprises of Wikipedia and USA Person. The created REST API will be utilized under two distinct circumstances, namely offline or local circumstances and online circumstances.

(b) The researcher develops a test scenario (experiment) after the REST API has been created in order to collect data. These are the possible outcomes:

   a To begin with, testing is done locally or offline on the REST API, which was created with Javascript and Golang utilizing Wikipedia and USA person data.

   b The REST API, which was created using Javascript and Golang and used Wikipedia and US person data, underwent both testing procedures online.

   c The section describing the data collection procedure will explain the process of collecting data in offline, local, and online settings.

(c) The author uses a browser with an inspect capability to receive data retrieval results and to view how much RAM is being used.

B Data gathering phases in both offline and online environments

(a) The author uses the inspect capability of the browser to run the test and collect the test results.

(b) Before creating the url to display the dataset from the REST API, the author uses the inspect capability in the browser.

(c) After launching the browsers inspect feature, the author selects the memory tab it owns and takes a screenshot to record the memory consumption.

(d) The author keeps track of the memory resources used while the dataset's data is displayed.

(e) The browser is closed and steps 1 through 5 in 33 times once the data has been entered into Excel.

(f) After performing the testing process 33 times, the author stopped using Javascript to run the REST API and turned the computer off for 10 to 15 minutes. Then, the computer was turned back on, and the author repeated steps 1 through 5 in 33 times using the Golang-developed REST API for US person data.
(g) After doing the testing process 33 times, the author stopped using the Golang REST API and left the laptop off for 10 to 15 minutes. Then, the laptop was put back on, and the author ran the Javascript REST API for Wikipedia data and repeated steps 1 through 5 in 33 times.
(h) After doing the testing procedure 33 times, the author stopped using the REST API made in Javascript and shut off the laptop for 10 to 15 minutes. Then, after turning it back on, the author used the REST API created in Golang and repeated steps 1 through 5 in 33 times.

C The method for gathering data in local or remote settings.
(a) The author operates a REST API for US person data that was created locally using Javascript.
(b) Carry out step B.

D The method for gathering data in an online environment.
(a) The author uploads the database and REST API to the hosting service that they employ so that a browser can access it online.
(b) The author manages the REST API, which was created online for USA person data using Javascript.
(c) Carry out step B.

2.7. Testing Process

In order to prevent errors in the testing process that will be executed later, the testing process will be carried out by adhering to the test scheme that has been developed previously.

2.8. Mean Value

Equation (1) displays the equation that was used to determine the average value. Where N is the number of samples that were examined, $\bar{x}$ is the average, and Σx is the average of the values of x added together [19].

$$\bar{x} = \frac{\Sigma x}{N} \tag{1}$$

2.9. Standard Deviation Value

Equation (2) shows the formula used to determine the standard deviation's value. Where n is the total number of samples tested, x_i is the value from x to i or x is the test value, and i is the value between 1 and 33, and x is the sample mean value [20].

$$s = \sqrt{\frac{\Sigma(x_i - \bar{x})}{n-1}} \tag{2}$$

2.10. Margin of Error Value

The equation used to find the value of the margin of error can be seen in equation (3). In this study, the critical importance is using a t-score or t-table. To determine the critical value required degrees of freedom and alpha level. For degrees of freedom or the number of samples used 33, while for the alpha value of 5%, where it is found that the t value or critical value is 2.0369 [21].

$$\text{margin of error = critical value}*(\frac{\text{standard deviation}}{\sqrt{n}}) \tag{3}$$

2.11. Equivalence Test

In the equivalence test, there are four hypotheses: the first is equivalent and not different, the second is not equivalent and different, the third is equivalent and different, and the fourth is not equivalent and not different. The equivalence test is a hypothesis test that is used to draw statistical conclusions from the data being observed from the results of the tests performed [22].

2.12. Normality Test

The Kolmogorov-Smirnov normality test can be employed in the equation to determine if the data is normally distributed or not [23], equation (4) displays the used formula.

$$D_n = sup|F_n(x) - F(x)| \tag{4}$$

2.13. T Test

One of the hypothesis testing methods that is not independent and is characterized by the existence of a value association in each of the same samples is the T-test for paired data (pairs). A normality test must be performed, and the data being examined using the t test for paired data must be in a state where it is normally distributed, according to the conditions for testing the T test for paired data [24], Equation (5) displays the equations that were employed.

$$t_{hit} = \frac{\bar{D}}{\frac{SD}{\sqrt{n}}} \tag{5}$$

2.14. Wilcoxon Test

To ascertain if the means of two paired samples differ, the Wilcoxon test is utilized. If the sample data are not normally distributed, the Wilcoxon difference test is performed [25], Equation (6) displays the used formula.

$$z = \frac{MAX(W^-, W^+) - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24} - \frac{t^3 - t}{48}}} \tag{6}$$

2.15. Research Tools

Hardware (such as laptops and hosting) and software (such as web servers, browsers, database engines, and others) were both used as research tools in this study. Table 3 lists the parameters of the laptop utilized for this study.

Table 3. Laptop and hosting specifications

Description	Laptop Specifications	Hosting Specifications
Brand / Type	Vaio / SVF14A15SGB	-
Processor	Intel Core i3-3337U	1 Core
Memory	8GB DDR3	765MB
Storage	SSD 240GB	SSD 2GB
Operating system	Windows 10 Pro 64-bit	-
IO	-	76800kBps

The used laptop's specifications are higher than the hosting's specifications. It is sufficient to run Javascript or Golang REST APIs with the hosting requirements indicated in table 3.

Table 4. Specifications for browsers, database engines, web servers, and programming languages

Description	Browser Specifications	Database Engine Specifications	Web Server Specifications	Programming Language Specification
Name	Mozila Firefox	MariaDB	Apache	Nodejs and Golang
Version	94.02 (64-bit)	10.4.13-MariaDB	Apache/2.4.43(Win64)	Nodejs-14.18.2 and Golang-1.17.5

Table 4 displays the requirements for testing, which include the Mozilla Firefox web browser, MariaDB as the database engine, Apache as the web server, and Javascript using node.js with version 14.18.2 and Golang with version 1.17.5.

Equation (7) is used to compare how differently Javascript and Golang use memory resources on average in Tables 3 and 4.

$$\text{average difference} = \text{biggest value average JS or GO - smallest value average JS or GO} \tag{7}$$

For each circumstance, the biggest or least average value from USA Person and Wikipedia is used in the aforementioned equation. Equation (8) shows the calculation that was done to determine what percentage of the difference in memory usage by using a laptop and hosting after obtaining the average difference in the use of memory resources between Javascript and Golang.

$$\text{memory usage} = \frac{\text{the smallest or largest value of the average difference}}{\text{device memory (laptop and hosting)}} \tag{8}$$

The internet used for this investigation is shown in Table 5, where it was provided by Indosat Ooredoo, with a download speed of 1.88 Mbps and an upload speed of 1.00 Mbps.

Table 5. Internet bandwidth

Description	Information
Provider Name	Indosat Ooredoo
Download	1.88Mbps
Upload	1.00Mbps

3. RESULT AND ANALYSIS

Three results were obtained, including the mean, standard deviation, and margin of error of memory resource usage, based on testing on the REST API that was done 33 times on each technology using two datasets made up of USA Person and Wikipedia. The tests were done under two different conditions, namely offline or local conditions and online conditions.

Each REST API created using Javascript and Golang, two distinct technologies, has its memory use monitored. The test's findings will be reported in the form of average, standard deviation, and error margin calculations.

For the first test, which was conducted offline utilizing the same two data, namely USA Person and Wikipedia, two conditions were tested: offline and online. To determine the average memory resource utilization of two alternative technologies, their respective standard deviations, and their margins of error.

Table 6. Memory condition offline

Technology	Average		Standard Deviation		Margin of Error	
	Usa person	Wikipedia	Usa perso	Wikipedia	Usa person	Wikipedia
Javascript	18,94MB	41,29MB	0,015	0,006	0,005	0,002
Golang	21,54MB	44,23MB	0,020	0,012	0,007	0,004

The first test used the same two data, USA Person and Wikipedia, and was conducted offline. Table 6 provides the average, standard deviation, and margin of error for the consumption of memory resources across two categories of technology.

Table 7. Memory condition online

Technology	Average		Standard Deviation		Margin of Error	
	Usa person	Wikipedia	Usa person	Wikipedia	Usa person	Wikipedia
Javascript	18,93MB	41,29MB	0,013	0,022	0,005	0,008
Golang	21,50MB	41,36MB	0,355	6,331	0,126	2,245

The same two data, USA Person and Wikipedia, were used in the second test, which was conducted online. Table 7 provides the average, standard deviation, and margin of error for the consumption of memory resources across two different technologies.

The results of the average calculation reveal that Javascript and Golang use memory resources differently on average. Using the equivalency test, t-test for paired data, and Wilcoxon test, a statistical study was done to back up this claim.

The following are the hypotheses for the Wilcoxon test and the t-test for paired data:

H0	: There is no difference between Javascript and Golang in how much memory they use on average.
H1	: There is a difference between Javascript and Golang in how much memory they use on average.
H0	: will not be accepted as the result of the test if the significance level is less than 0.05.

3.1. Equivalence Test

The equivalency test is run to generate statistical inferences from the data collected throughout the test. The average value, standard deviation, and margin of error calculation results are plotted on a graph for the equivalency test. There will be two forms of the graph representing the equivalence of the average value of the data. The average memory consumption for Javascript and Golang is shown in Figure A, along with each average's standard deviation. Figure B shows the standard deviation of the opposing technology and the average memory usage for Javascript and Golang.

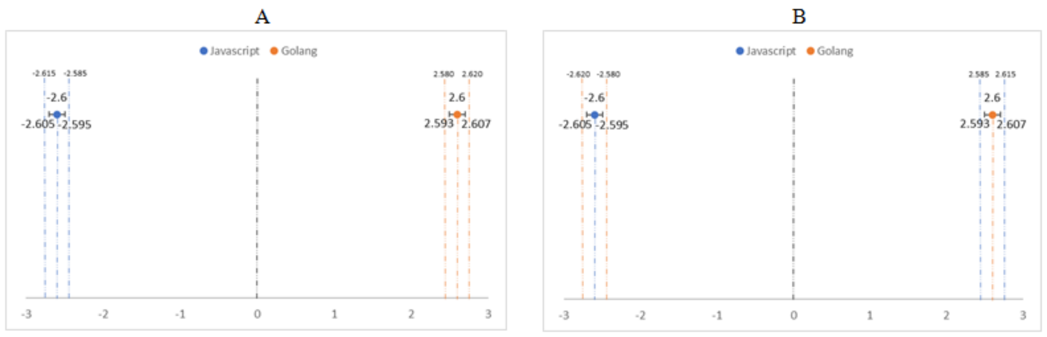

Figure 3. Graph of average memory usage in the case of loading USA person data which is carried out offline

If an equivalence test is run based on the graph in Figure 3, it can be shown that Javascript and Golang use memory resources differently on average while testing with USA Person data offline.

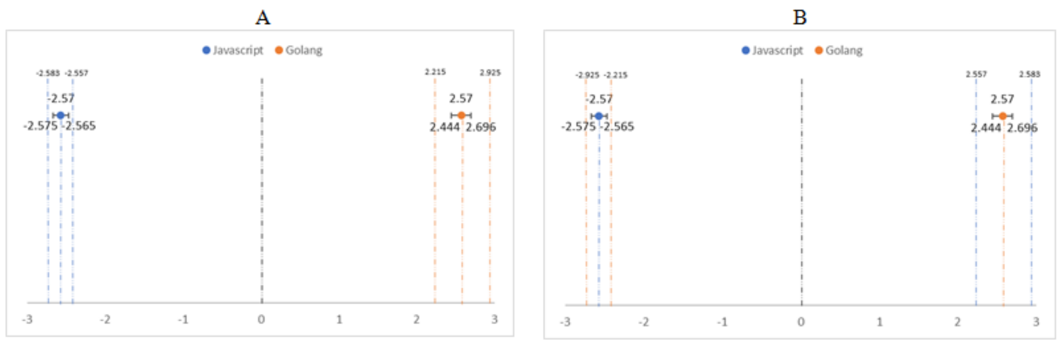

Figure 4. Graph of average memory usage in the case of loading USA person data which is carried out online

If an equivalence test is conducted using data from USA Persons online, it can be shown from the graph in Figure 4 that Javascript and Golang use memory resources differently on average.

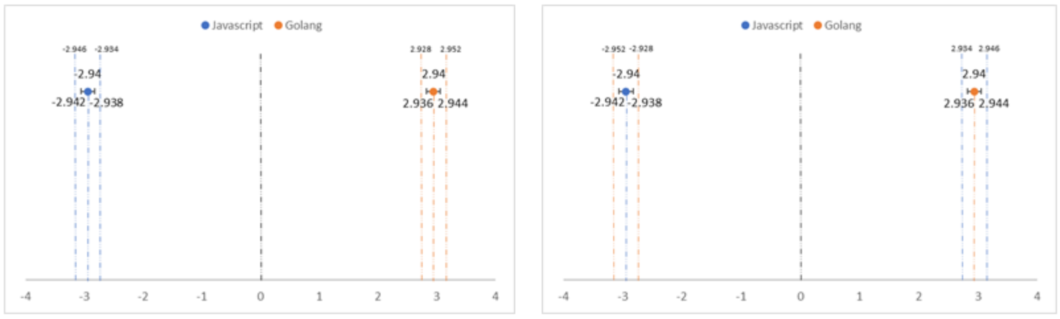

Figure 5. Graph of average memory usage in the case of loading Wikipedia data which is carried out offline

If an equivalence test is conducted with offline Wikipedia data, it can be shown from the graph in Figure 5 that Javascript and Golang use memory resources differently on average when testing.

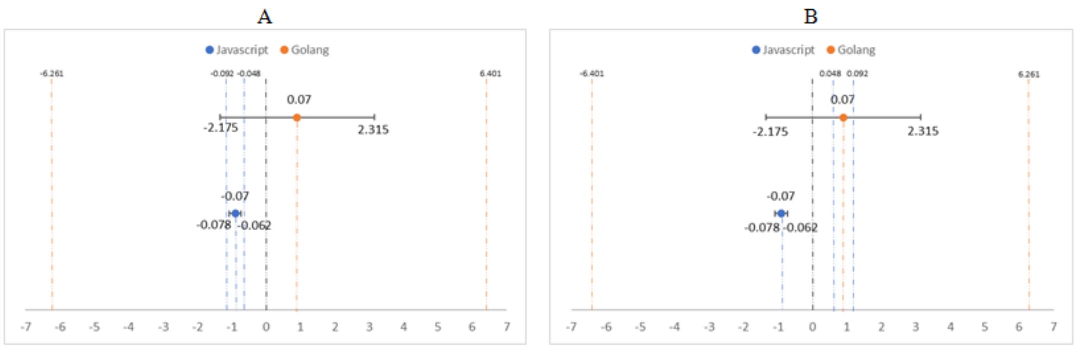

Figure 6. Graph of average memory usage in the case of loading Wikipedia data which is carried out online

If an equivalence test is conducted using online Wikipedia data, it can be observed from the graph in Figure 6 that there is no difference between Javascript and Golang in terms of the average use of memory resources.

3.2. T Test

The t-test for paired data was used for statistical analysis, with the underlying assumption that the data being examined were normally distributed. As a result, the data to be examined are subjected to a normality test before the t-test for paired data is run.

Table 8. Test results for normality of memory usage in the case of load USA person and Wikipedia data offline

Technology	USA person	Wikipedia
JavaScript	0.012	0.000
Golang	0.000	0.000

Table 9. Test results for normality of memory usage in the case of load USA person and Wikipedia data online

Technology	USA person	Wikipedia
JavaScript	0.001	0.000
Golang	0.000	0.000

The statistics on the memory use of the Javascript and Golang application prototypes were evaluated with USA Person and Wikipedia data that were carried out offline and online were found to be non-normally distributed based on the Kolmogorov-Smirnov normality test in Tables 8 and 9.

The t-test for paired data was nevertheless used in this investigation even though it appears from the results of the normality test that the data are not regularly distributed. The central limit theorem, which asserts that for large numbers of data (more than 31), the data can still be deemed normally distributed even though the results of the normality test are not normally distributed, is used to support this claim.

Table 10. The results of the t-test for paired data on memory usage in the case of offline USA Person and Wikipedia data load

Technology	Sig
USA person	
JavaScript dan Golang	0.000
Wikipedia	
JavaScript dan Golang	0.000

Thirty-three times were run the paired data t-test with an alpha level of 0.05 and a statistical power of 0.999 (close to 1). Table 10's paired data t-test findings showed a significant value of 0.000, which is less than 0.05. Therefore, H0 is disregarded and H1 is accepted; as a result, statistically, Javascript and Golang use memory resources differently on average.

Table 11. The results of the t-test for paired data on memory usage in the case of online USA Person and Wikipedia data load

Technology	**Sig**
Usa person	
JavaScript dan Golang	0.000
Wikipedia	
JavaScript dan Golang	0.953

When the application prototype was tested using USA Person online with a significance level less than 0.05, the results of the t-test for paired data as presented in table 11 obtained a significant value of 0.000. As a result, H0 is disproved and H1 is accepted, indicating that statistically, Javascript and Golang use memory resources differently on average. The significance value for the t-test for paired data on the application prototype tested utilizing online Wikipedia was 0.953 more than 0.05. Because H1 is disregarded and H0 is accepted, there is no statistically significant difference between Javascript and Golang's average utilization of memory resources.

3.3. Wilcoxon Test

It was discovered that the data were not normally distributed based on the normality test on the average data utilization of memory resources between Javascript and Golang. Therefore, the Wilcoxon test will be used to determine whether the data are different.

Table 12. Wilcoxon test results on memory usage on offline USA Person and Wikipedia data loads

Technology	**Sig**
Usa person	
Java Script dan Golang	0.000
Wikipedia	
JavaScript dan Golang	0.000

Table 13. Wilcoxon test results on memory usage on online USA Person and Wikipedia data loads

Technology	**Sig**
Usa person	
JavaScript dan Golang	0.000
Wikipedia	
JavaScript dan Golang	0.079

The Wilcoxon test was used to analyze the results, which yielded a significant value of 0.000 for the offline USA Person and Wikipedia data in table 12 and the online USA Person data in table 13. As a result, H1 is approved and H0 is refused. Therefore, statistically, Javascript and Golang use memory resources differently on average. Table 13's results for the online Wikipedia data obtained a significance value of 0.079. As a result, H0 is approved and H1 is refused. In order for Javascript and Golang to consume memory resources on average similarly, statistically.

3.4. Result Analysis

Based on the equivalency test, t-test for paired data, and Wilcoxon test, it was discovered that Javascript and Golang used memory resources differently on average for three of the four trials that were undertaken. However, while comparing the average use of memory resources between Javascript and Golang when evaluating the Wikipedia data online, it was discovered that there was no difference based on the equivalence test, t-test for paired data, and Wilcoxon test. The change is essentially minimal, as evidenced by the comparatively small difference in average memory resource utilization (practically insignificant). According to this study, the average memory usage of Javascript and Golang differs only 0.07MB to 2.94MB, meaning that the difference occupies just 0.0085% to 0.036% of laptop memory (8GB) and 0.0092% to 0.388% of server memory (765MB). Practically, this extremely slight change appears to have no effect on how memory resources are used.

4. CONCLUSION

The results of the equivalence test, t-test for paired data, and Wilcoxon test indicate that Javascript and Golang use memory resources differently, but the difference is essentially inconsequential (practically significant).

It is important to carry out additional research through trials utilizing a bigger sample of data objects or with alternative test scenarios in order to confirm the findings of this study.

5. ACKNOWLEDGEMENTS

I'd like to express my gratitude to the State Islamic University of Sunan Kalijaga and the Journal Matrix manager for giving me the chance to contribute to the Journal's publishing.

6. DECLARATIONS

AUTHOR CONTIBUTION

The first author carried out the experiment and wrote the manuscript with support from the second author. The second author conceived the idea, develop the theory, verified the analytical method and supervised the findings of this work. All authors discussed the result and contributed to the final manuscript.

FUNDING STATEMENT

The research leading to these results received no funding from any parties.

COMPETING INTEREST

The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work.

REFERENCES

[1] J. Duda and W. Dlubacz, "IMPROVING EFFICIENCY OF AWEB-BASEDDISTRIBUTED EVOLUTIONARY IMPROVING EFFICIENCY OF A WEB-BASED," pp. 143–150, 2019.

[2] O. S. Akanji, O. A. Abisoye, and M. A. Iliyasu, "Mitigating Slow Hypertext Transfer Protocol Distributed Denial of Service Attacks in Software Defined Networks," *Journal of Information and Communication Technology*, vol. 20, no. 3, pp. 277–304, 2021.

[3] P. J. Roig, S. Alcaraz, K. Giily, and C. Juiz, "Algebraic Formal Modelling for HTTP Main Methods using ACP," pp. 1–6, 2019.

[4] B. Chinthanet, S. E. Ponta, H. Plate, A. Sabetta, R. G. Kula, T. Ishio, and K. Matsumoto, “Code-based vulnerability detection in Node.js applications: how far are we?” vol. 20, pp. 1199–1203, 2020.

[5] J. Whitney, C. Gifford, and M. Pantoja, “Distributed execution of communicating sequential process-style concurrency: Golang case study,” vol. 75, pp. 1396–1409, 2019.

[6] N. N. Joseph, R. N. Roy, and T. A. Steitz, “pdbmine: A Node.js API for the RCSB Protein Data Bank (PDB),” pp. 1–4, 2019.

[7] T. Turc, “Internet of Things Based on Http,” *Scientific Bulletin of the Petru Maior University of Targu Mures*, vol. 15, no. 2, pp. 5–8, 2019.

[8] M. Bavdys, “Golang Multithreading,” no. 10, pp. 38–40, 2108.

[9] S. Sukaridhoto, D. K. Basuki, H. Yulianus, and R. P. N. Budiarti, “Performance Evaluation of Integrated Deep Learning Web Platform for Dataset Training,” *Applied Technology and Computing Science Journal*, vol. 2, no. 2, pp. 117–128, 2020.

[10] S. S. Brimzhanova, S. K. Atanov, M. Khuralay, K. S. Kobelekov, and L. G. Gagarina, “Cross-platform compilation of programming language Golang for Raspberry Pi,” *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, vol. 19, no. 10, pp. 1–5, 2019.

[11] A. A. Kristanto, Y. Harjoseputro, and J. E. Samodra, “Implementasi Golang dan New Simple Queue pada Sistem Sandbox Pihak Ketiga Berbasis REST API,” vol. 1, no. 10, pp. 7–8, 2021.

[12] H. Brar, T. Kaur, and Y. Rajoria, “The Better Comparison between PHP , Python-web & Node . js,” vol. 9, no. 7, pp. 29–37, 2021.

[13] H. K. Dhalla, “A Performance Comparison of RSTful Applications Implemented in Spring Boot Java and MS.Net Core,” vol. 1933, pp. 1–7, 2020.

[14] F. Effendy, T. Taufik, and B. Adhilaksono, “Performance Comparison of Web Backend and Database: A Case Study of Node.JS, Golang and MySQL, Mongo DB,” vol. 14, no. 14, pp. 1955–1961, 2021.

[15] A. A. Pangera and D. Ariyus, “Manajemen Memory,” *STMIK AMIKOM Yogyakarta, Teknik Informatika*, pp. 1–6, 2017.

[16] D. Clark and S. Wedeman, “Measurement , Meaning and Purpose : Exploring the M-Lab NDT Dataset,” pp. 1–44, 2021.

[17] A. Apriani, H. Zakiyudin, and K. Marzuki, “Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF System Penerimaan Mahasiswa Baru pada Kampus Swasta,” *Jurnal Bumigora Information Technology (BITe)*, vol. 3, no. 1, pp. 19–27, 2021.

[18] A. Arcuri, “RESTful API automated test case generation with Evomaster,” *ACM Transactions on Software Engineering and Methodology*, vol. 28, no. 1, pp. 1–37, 2019.

[19] J. Shi, D. Luo, H. Weng, X. T. Zeng, L. Lin, H. Chu, and T. Tong, “Optimally estimating the sample standard deviation from the five-number summary,” *Research Synthesis Methods*, vol. 11, no. 5, pp. 641–654, 2020.

[20] S. Harrisson, “The downside of dispersity: Why the standard deviation is a better measure of dispersion in precision polymerization,” *Polymer Chemistry*, vol. 9, no. 12, pp. 1366–1370, 2018.

[21] S. Wicket, “Margin of error,” *Journal of Cell Science*, vol. 132, no. 15, pp. 1–2, 2019.

[22] M. Hasyim, P. Kuswarini, and Kaharuddin, “Semiotic Model for Equivalence and Non-Equivalence in Translation,” *Humanities & Social Sciences Reviews*, vol. 8, no. 3, pp. 381–391, 2020.

[23] D. S. Dimitrova, V. K. Kaishev, and S. Tan, “Computing the kolmogorov-smirnov distribution when the underlying cdf is purely discrete, mixed, or continuous,” *Journal of Statistical Software*, vol. 95, no. 10, pp. 1–42, 2020.

[24] C. Montolalu and Y. Langi, “Pengaruh Pelatihan Dasar Komputer dan Teknologi Informasi bagi Guru-Guru dengan Uji-T Berpasangan (Paired Sample T-Test),” *d’CARTESIAN*, vol. 7, no. 1, p. 44, 2018.

[25] M. Bellaiche, R. Oozeer, G. Gerardi-Temporel, C. Faure, and Y. Vandenplas, “Multiple functional gastrointestinal disorders are frequent in formula-fed infants and decrease their quality of life,” *National Library of Medicine*, pp. 1276–1282, 2018.