

Performance Testing and Optimization of DiTenun Website

Arlinta Christy Barus*, Eka Stephani Sinambela, Ivani Purba, Jhonathan Simatupang,
Monika Marpaung, & Nancy Pandjaitan

Faculty of Informatics and Electrical Engineering, Institut Teknologi Del, 22381, Laguboti, Indonesia

Abstract

DiTenun website is a web-based application developed to create new motifs or new weaving patterns. This website needs to have good performance so that it can be optimally used by the users. By understanding the importance of the information available on the DiTenun website, a performance testing is necessary. This performance test aims to verify specific system performance such as response time and service availability despite receiving a large number of requests. The performance analysis of the DiTenun website was carried out by spike testing using the Apache JMeter tool. In solving the problems and improving the performance of the DiTenun website, the researcher applied website optimization by utilizing the browser cache, activating gzip compression, and eliminating rendering – blocking resources. Evaluation of each solution was carried out by using a hypothesis test, namely using a paired sample t test. The result of the test showed that each solution that was implemented can be used to improve performance in the form of load time of the DiTenun website.

Keywords: DiTenun Website, Performance, Load Time, Optimization

Received: 15 March 2022

Revised: 29 May 2022

Accepted: 2 June 2022

1. Introduction

Every software is built with the hope that it can be used by users according to its purpose. A software should have good performance so that it can be used comfortably by users. Performance is the most important factor in determining the quality of a software, whether it is being or has been built. The performance in question refers to the ability of the system to respond to information transactions better, faster, and more accurately even though the system is being accessed by more than one user at the same time (multi-user) (Adrian & Jason, 2020). This is what underlies the need for testing the performance of an application, especially web-based applications. The test in question is to obtain information about the performance specifications of the software to be tested.

The software that is the object of this research is the DiTenun website application which focuses on developing Ulos motifs. DiTenun focuses on developing Ulos motifs automatically using machine learning technology and concepts by using an image processing algorithm that learns about Ulos motifs and produces new Ulos motifs but still has the cultural values contained in Ulos itself (Putra et al., 2020).

By understanding the importance of the information available on the DiTenun website to the wider community, the researchers felt the need to perform performance testing so that this device can be used comfortably by its users. This performance test aims to verify specific system performance such as response time and service availability even when receiving a large number of requests. After testing the performance of the DiTenun website, researchers are expected to be able to optimize aspects that affect response time and service availability on the DiTenun website, so that it can be used more comfortably by users. The process of analyzing the performance of the DiTenun website is carried out by applying a spike test. (Sha et al., 2010).

* Corresponding author.

E-mail address: arlinta@del.ac.id

2. Literature Study

2.1 DiTenun Website

DiTenun website is a web-based application developed to create new motifs or new weaving patterns. The motifs are made using variations of motifs from existing traditional weaving and even new motifs produced by the DiTenun software. The DiTenun website itself examines machine learning systems to classify Nusantara weaving, generate new motifs quickly and automatically, preserve Indonesian culture and make it universally accessible.

2.2 Performance Testing

Performance testing is a type of testing to ensure the software will work well under the expected workload (Permatasari, 2020). The focus on the performance test itself is divided into several parts, namely:

- Speed – To determine the response speed of the application when receiving a request or action.
- Scalability – To determine the maximum amount of user load that the application can handle.
- Stability – To determine the stability of the application under various loads.

2.3 Spike Test

Spike testing is the process of adding a sudden load and analysing its behaviour, whether to reduce or manage load changes. Spike testing is done by increasing the number of users suddenly by a very large number and measuring system performance. Spike testing aims to determine the ability of a system to maintain workloads. When doing spike testing, application performance can slow down, deteriorate, or even stop. When that happens, the Tester can find the point of failure of the application. The figure 1 show the steps in performing spike testing.

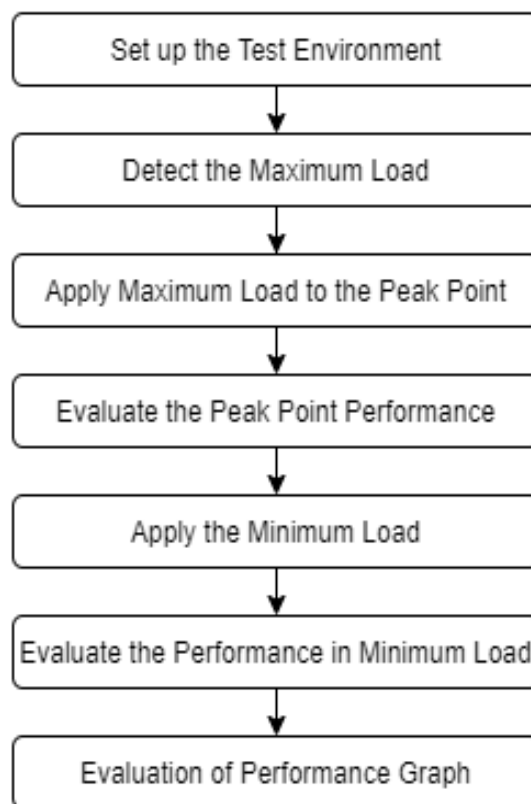


Figure 1. Spike Test Process Diagram

The approach taken in conducting spike testing consists of 7 steps as shown in Figure 1.

(1) Set up the test environment

The first step in performing a spike test is to prepare a testing environment with the aim that the tests carried out can be successful. At this stage, it will also be ensured that no one spikes the application to be tested at the same time.

(2) Detect the Maximum Load

After setting up the test environment, the next step is to determine the maximum load of the application to be tested. The maximum load in question is the maximum number of users who can access the application simultaneously without being found an error.

(3) Apply Maximum Load to the Peak Point

The next step is to increase the load unexpectedly at a certain time, then give a predetermined maximum load to the peak point using the help of tools.

(4) Evaluate the Peak Point Performance

The next stage is to evaluate the maximum load increase performance at the peak point to check whether the system will fail or survive under an unexpectedly increased load.

(5) Apply the Minimum Load

At this stage, the load will be lowered from the highest value to the lowest value (minimum) unexpectedly at a certain time using the help of tools.

(6) Evaluate the Performance in Minimum Load

In this step, we will evaluate performance under minimum load to see if the system will survive or fail under unexpectedly decreasing load.

(7) Evaluation of Performance Graph

The final step is to analyze the performance graph by looking at the metrics generated from the test.

2.4 Front-end

Front-end is a technology used in building websites from the client side so that users can see and interact directly with the application. The front-end is built to ensure that when a user opens a website, the user can view information in a format that is easy to read and understand (Ridwan, 2020). The front-end is built using HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. HTML, CSS, and Javascript are components that make a URL work and display an attractive website.

2.5 The Apache JMeter

The Apache JMeter™ is a Java-based website analysis application that can run on any operating system and is able to test the stability and performance of a website (Chandra, 2019). There are several elements or terms in JMeter that can be used to help understanding. The following guides in understanding each input and output of testing using the JMeter tools.

- (1) Samples, providing information on the number of requests created to be sent to the server.
- (2) Average, provides information about the average elapsed time obtained from requests sent to the server.
- (3) Elapsed time, the time from the moment before sending the request to just after the last response was received.
- (4) Throughput, provides information about the number of requests processed per second. The Value limit throughput depends on the maximum value bandwidth.
- (5) Error Rate, provides information about the percentage of errors that occur on the request.
- (6) Initial Delay, the waiting time required before executing a thread from a certain line.
- (7) Startup Time, the upgrade time for all threads specified on the number of threads. For example, for 10 threads, the startup time is set at 10 seconds, then 1 will appear thread new every 1 second.

- (8) Hold Load, the duration to maintain a specified number of concurrent users and is completely independent of the upgrade time.
Shutdown Time, the time interval for deactivating all threads active.

2.6 WebPageTest

WebPageTest is a tool used to test the performance of a website if it is accessed by many people at once. The result is a time that is divided into several parts, namely the time when DNS lookup, TCP connection time, HTTP response time and others. In addition to time, bandwidth is also measured during the website testing process (FIRDAUS et al., 2019). WebPageTest is able to provide test results that are presented specifically and in detail so that the data can be used by Examiners to study the advantages and disadvantages or aspects that must be improved from the website being tested.

2.7 W3 Total Cache

W3 Total Cache is a WordPress plugin that functions to optimize the performance or performance of a website. This plugin is the only Web Performance Optimization (WPO) framework for WordPress that has been trusted by millions of publishers, web developers, and web hosts around the world for more than a decade (Gunawan & Saputro, 2018).

3. Analysis and Methodology

3.1. DiTenun Website Analysis

The object of this study is DiTenun website. DiTenun is a smart device developed as a digital transformation initiative for Nusantara weaving motifs. Based on the type of user, DiTenun is divided into two, namely DiTenun – Web App and DiTenun – Commercial Web. DiTenun web app is designed for weavers to generate motifs, as a critical editor, and gallery management. DiTenun commercial web is designed as a website application for managing and buying and selling products.

3.2. Front-end Website Analysis

Here are some aspects from the side front-end that can affect the performance of a website.

a. Browser Caching

A website takes a long time to load large files and if the internet connection is slow it will slow down the process of loading website pages. Each file makes a separate request to the server, the more requests that are sent to the server at the same time the more work that must be done so that it can reduce website speed. Some files on the DiTenun website can be stored in a local browser by using browser caching. So on the DiTenun website it is necessary to enable browser caching so that the loading time of website pages becomes shorter. (Adrian & Jason, 2020)

b. Compression

When a user accesses a web page, a request is sent to the server to provide the requested file. The larger the file on the website, the longer it will take to load the web page. Based on the test results obtained, the files on the DiTenun website can still be compressed to make them smaller in size. So that the time needed to load DiTenun's website pages also becomes faster.

c. Render-Blocking Resource

When a user accesses a web page, he or she needs to download or render all the content on the web page, including HTML, Javascript, and CSS code. Render-Blocking is an activity to stop a page from appearing until the user has finished loading it, then what appears to the user is a blank white screen. (Putri et al., 2017)

When discussing render-blocking resources, we usually refer to Javascript and CSS. However, not all Javascript and CSS files perform rendering-blocking resources. CSS affects the model of website pages, positioning elements, determining layouts, and making websites responsive. So without CSS, the website will just look like a big block of text with some images and links. Javascript can create dynamic websites and change their

appearance quickly. There are several files that can render blocking resources on the DiTenun website so it is necessary to eliminate or delete files that can render blocking resources. (Saputra, 2012)

3.3. Experimental Design

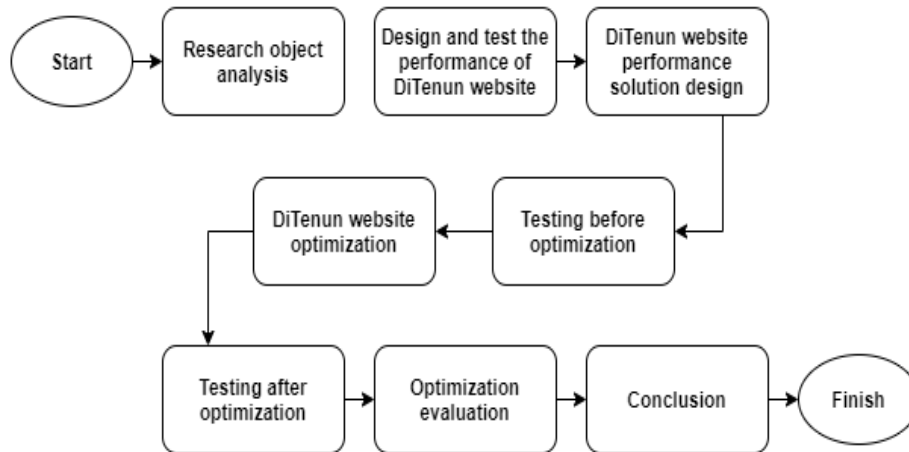


Figure 2. Experimental Research Design

Figure 2 is an experimental design carried out by researchers in carrying out this research.

(1) Research object analysis

This stage was describing the details of the object used in the study. The object used in this research is the DiTenun website. The analysis was carried out on aspects of the DiTenun website that can affect website performance.

(2) Designing and test the performance of DiTenun website.

This stage was the stage of designing and implementing tests carried out by researchers. The design of the test included identifying the needs in the testing process to be carried out. The tests carried out were spike tests on the DiTenun web app and commercial web using JMeter tools to observe the behavior of the website under a sudden increase and decrease in load.

(3) DiTenun Website Performance Solution Design

In this stage, researchers conducted the design of solutions that optimize commercial web performance based on the problems obtained in the testing that has been done before.

(4) Testing before optimization

At this stage, the researcher tested the DiTenun commercial web using WebPageTest to get the load time on each page. The load time in this test was compared with the test load time after optimization to evaluate the given solution.

(5) DiTenun website optimization

This stage was the stage of optimizing DiTenun's commercial web according to the previously designed performance solution.

(6) Testing after optimization

At this stage, the researcher tested the DiTenun commercial web using WebPageTest to get the load time on each page. The load time in this test was compared with the test load time before optimization to evaluate the given solution.

(7) Optimization evaluation

At this stage, the researcher conducted an evaluation using statistical testing of the DiTenun commercial web load time data before and after the optimization was carried out.

(8) Conclusion

At this stage, the researcher draws conclusions after carrying out each stage of the research and draws conclusions based on the results of the analysis.

3.4. Design Spike Test

The whole process of doing spike testing was divided into several stages which can be seen in the diagram Figure 1. Based on the diagram of the process, the following was the experimental design of the spike test on the DiTenun website.

(1) Set up the test environment

This stage was the stage of determining the test environment that was used to perform spike testing. Metrics observed during testing included error rate and elapsed time under a sudden increase and decrease in load using the JMeter tool. The scenario determined in this study was considered to have covered all the activities to be carried out on the website.

(2) Detect the Maximum Load

The table 1 show the maximum load of DiTenun website users.

Table 1. Maximum User Limit of DiTenun Website

No.	Trial	Number of Users
1.	TC-01 Beranda	140 User
2.	TC-02 Katalog Produk	133 User
3.	TC-03 Keranjang	133 User
4.	TC-04 Penenun Gedongan	132 User
5.	TC-05 Swadesain	133 User
6.	TC-06 Riset	133 User
7.	TC-07 Login Webapp	132 User
8.	TC-08 Gambar Kristik	132 User
9.	TC-09 Hasilkan Kain	133 User
10.	TC-10 Motif Saya	133 User
11.	TC-11 Kain Saya	133 User
12.	TC-12 Akun	132 User
13.	TC-13 Hasilkan Motif	133 User

(3) Apply Maximum Load to the Peak Point

At this stage, the Examiner applied the maximum load that has been obtained previously on the website with the help of JMeter tools for each test case.

(4) Evaluate the Peak Point Performance

At this stage, the Tester evaluated the performance and behavior of the system at peak or under maximum load.

(5) Apply the Minimum Load

At this stage, the Tester applied the minimum load on the website using JMeter tools for each test case.

(6) Evaluate the Performance in Minimum Load

At this stage, the Tester evaluated the performance and behavior of the system under minimum load.

(7) Evaluation of Performance Graph

At this stage, the examiner evaluated the test results from the resulting metrics. The behavior of the system was observed when given load increased and decreased. Observations were also made on the system

recovery process to stabilize and return to normal function after the load was increased to the maximum point. Test analysis was carried out based on elapsed time and error rate.

3.5. Designing Optimization Solutions

Performance optimization from the front-end side is an important thing to do to create a good experience for website users. Here are some things that can be done to optimize the performance of DiTenun commercial web.

(1) Utilizing Cache Browser

Utilizing browser caching can save some files locally in the user's browser. A user's first visit to a web page will take up the same loading time. But on subsequent visits or when the user visits another page on the website, the browser already has some of the files it needs locally. This causes the amount of data downloaded by the browser to be less, so that the loading time of web pages will be reduced. The way that can be done to activate browser caching on the DiTenun website is to activate the W3 Total Cache plugin.

(2) Enabling Gzip Compression

Enabling gzip compression can help reduce the size of HTML, CSS, Javascript, json, xml, and font files on DiTenun websites. HTML and CSS files generally use repeated text and spaces, so gzip compression can compress page and CSS sizes by up to 60%. Gzip compression can be enabled by using the W3 Total Cache plugin.

(3) Eliminated Render – Blocking Resource

Eliminate render-blocking resources can affect website page loading times to be faster and maximum. The way that can be done to eliminate rendering-blocking resources on the website is to use a plugin. A plugin that can be used to eliminate rendering – blocking resources is W3 Total Cache. With the help of plugins, rendering-blocking elimination can be done on CSS and Javascript files.

4. Results and Discussion

4.1. Performance Testing (Spike Test)

The table 2 show the results of Testing the performance of the DiTenun website by applying the method spike test.

Based on table 2, the average load time for all test cases is 6.1 seconds where the test results of each test case has an error percentage rate above 0%. Although each test case has a percentage error rate, on several pages, namely Kristik Editor (TC-SP08), Layout Sadum (TC-SP09), and Hasilkan Motif (TC-SP13) page, an error rate of 0% is obtained, which means that the page is can provide a successful response when there is a sudden increase and decrease in requests.

Based on table 2, overall it can be concluded that the DiTenun website is not good enough to respond to requests that increase and decrease suddenly. However, the error occurs when the active thread increases and the error slowly disappears after a while and the website can respond to subsequent requests well. So, based on this, it can be concluded that when an error occurs due to a sudden increase and decrease in load, the DiTenun website can slowly recover and return to normal by itself after some time without involving technicians.

4.2. Optimization Evaluation

The implementation of solutions was carried out by researchers to increase the load time of the performance of the DiTenun website:

(1) Utilizing Cache Browser

Output results before and after the implementation of the browser cache can be seen in Table 3.

Load time before the implementation of the browser cache is longer than the load time after the implementation of the browser cache. Thus, the application of a browser cache on the website can improve the performance of the DiTenun website.

Table 2. Spike Test Result

Tast Case	Label	Samples	Total Samples	Average (ms)	Total Average (ms)	Error (%)	Total Error (%)
TC-SP01	Beranda	909	909	7975	7975	19.80	19.80
TC-SP02	Beranda	332	878	7495	7660	4.82	7.74
	Out Product	309		8685		9.71	
	Shop	237		6554		9.28	
TC-SP03	Beranda	234	897	8523	7416	0.85	3.79
	Out Product	229		11530		3.06	
	Shop	265		8836		4.85	
	Product Boras Pati	95		2373		7.37	
	Product Boras Pati	89		1870		6.74	
	Cart	85		1970		4.71	
TC_SP04	Beranda	284	923	6954	7202	1.06	1.41
	Artisans	276		9904		1.45	
	Penenun	224		7688		1.34	
	Gedongan	139		1562		2.16	
	Asti Panjaitan						
TC_SP05	Beranda	506	918	7840	7357	3.16	4.58
	Design Your Own	412		6764		6.31	
TC_SP06	Beranda	292	754	8073	9556	0.68	1.86
	About	279		12644		2.87	
	Revitalisasi	183		7214		2.19	
	Tradisi Tenun						
TC_SP07	Beranda	445	1293	4972	4648	2.47	3.33
	Webapp	428		4643		3.74	
	Webapp Login	420		4310		3.81	
TC_SP08	Beranda	426	1648	5095	3765	6.93	8.62
	Webapp	453		4790		12.14	
	Webapp Login	424		3757		12.97	
	Kristik Editor	309		283		0.00	
TC_SP09	Beranda	429	1815	5174	3392	9.79	11.46
	Webapp	421		4431		14.01	
	Webapp Login	405		4353		14.57	
	My Layouts	284		1031		16.90	
	Create	276		59		0.00	
	Layot Sadum						
TC_SP010	Beranda	413	1451	5831	4330	3.15	6.96
	Webapp	396		5100		8.08	
	Webapp Login	280		4305		8.42	
	My Gallery	262		836		9.16	
TC_SP011	Beranda	412	1464	5537	4209	2.43	4.30
	Webapp	400		4937		4.75	
	Webapp Login	388		4390		4.90	
	My Layouts	264		770		5.68	
TC_SP012	Beranda	293	1405	5183	4191	0.51	1.78
	Webapp	385		5165		2.08	
	Webapp Login	376		4393		2.13	
	Settings	251		8400		2.79	
TC_SP013	Beranda	335	972	13233	7810	8.36	11.83
	Webapp	270		10601		15.93	
	Webapp Login	186		1525		23.66	
	Hasilkan Motif	181		70		0.00	

Table 3. Testing Results of Browser Cache Implementation

Test Data	Path	Load Time (Ms)		
		Before	After	Difference
Home page	https://ditenun.com/	6903	6778	129
Product Catalog	https://ditenun.com/shop/	4115	3957	158
Basket	https://ditenun.com/product/boraspati-card-holder/	5138	4939	199
Gedongan Weaver	https://ditenun.com/asti-paijatan/	4271	3905	366
Self-design	https://ditenun.com/desain-your-own/	6420	6104	316
Research	https://ditenun.com/revitalisasi-tradisi-tenun-lewat-desain-baru-ciptaan-Ditenun/	4178	3987	191

(2) Enabling Gzip Compression

Output results before and after implementing Gzip Compression can be seen in Table 4.

Table 4. Gzip Compression Implementation Test Results

Test Data	Path	Load Time (Ms)		
		Before	After	Difference
Home page	https://ditenun.com/	6903	6786	117
Product Catalog	https://ditenun.com/shop/	4115	3953	162
Basket	https://ditenun.com/product/boraspati-card-holder/	5138	4967	171
Gedongan Weaver	https://ditenun.com/asti-paijatan/	4271	3971	300
Self-design	https://ditenun.com/desain-your-own/	6420	6141	279
Research	https://ditenun.com/revitalisasi-tradisi-tenun-lewat-desain-baru-ciptaan-Ditenun/	4178	3984	194

Load time before implementing gzip compression is longer than the load time after implementing gzip compression. Thus, the application of Gzip Compression on the website can improve performance on the DiTenun website.

(3) Eliminated Render – Blocking Resource

Output results before and after implementation eliminated rendering – blocking resources can be seen in Table 5.

Table 5 Implementation Test Results Eliminated Render-Blocking Resource

Test Data	Path	Load Time (Ms)		
		Before	After	Difference
Home page	https://ditenun.com/	6903	6784	119
Product Catalog	https://ditenun.com/shop/	4115	3952	163
Basket	https://ditenun.com/product/boraspati-card-holder/	5138	5021	117
Gedongan Weaver	https://ditenun.com/asti-paijatan/	4271	3987	284
Self-design	https://ditenun.com/desain-your-own/	6420	6176	244
Research	https://ditenun.com/revitalisasi-tradisi-tenun-lewat-desain-baru-ciptaan-Ditenun/	4178	3987	200

Load time before implementation eliminated rendering – blocking resource is longer than load time after implementation of eliminated render – blocking resources. Thus, the application of eliminated render – blocking resources on websites can improve performance on the DiTenun website.

5. Conclusion

From the results obtained in this study, the researcher can conclude that:

- a. Performance testing with JMeter can provide a specific report on the performance of the website being tested in accordance with the tested metrics, namely elapsed time and error rate. Performance testing with JMeter can also provide user restrictions from the DiTenun website.
- b. The cause of availability problems and user inconvenience due to an error with response code 502 "bad gateway" when the user makes a request is the inability of the server to respond to many requests at one time because the server is overloaded.
- c. Load time optimization on the DiTenun website has been successfully carried out by implementing browser cache, gzip compression, and eliminating rendering - blocking resources. This solution has succeeded in reducing load time on the DiTenun website and has been proven statistically. After doing optimization there is a reduction in load time of 100 ms to 400 ms.

References

- Adrian, J., & Jason, H. M. S. (2020). Pengujian Performa Dan Tingkat Stres Pada Website Hotel di Kota Batam. *Jurnal Ilmiah Binary STMIK Bina Nusantara Jaya Lubuklinggau*, 2(2), 31–34.
- Chandra, A. Y. (2019). Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request. *Jurnal Sistem Dan Informatika (JSI)*, 14(1), 48–56.
- Firdaus, A. L. I., Widodo, S., Sutrisman, A. D. I., Gading, S., & Mardiana, R. (2019). Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Web Service Pada Jurusan Teknik Komputer Polstri. *INFORMANIKA*, 5(2).
- Gunawan, H., & Saputro, A. K. H. (2018). Pemanfaatan Aplikasi Mobile Untuk Mempercepat Pencarian Tempat Indekos Berbasis Android. *Jurnal Muara Sains, Teknologi, Kedokteran Dan Ilmu Kesehatan*, 1(2), 85–96.
- Permatasari, D. I. (2020). Pengujian aplikasi menggunakan metode load testing dengan apache jmeter pada sistem informasi pertanian. *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)*, 8(1), 135–139.
- Putra, D. A., Sasmita, G. M. A., & Wiranatha, A. A. K. A. C. (2020). E-Commerce Marketplace Petshop Menggunakan Integrasi Rajaongkir API dan iPaymu Payment Gateway API. *JITTER: Jurnal Ilmiah Teknologi Dan Komputer*, 1(1), 13–22.
- Putri, M. A., Hadi, H. N., & Ramdani, F. (2017). Performance testing analysis on web application: Study case student admission web system. *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, 1–5.
- Ridwan, A. (2020). Analisis Perbandingan Performa Apache Web Server dan Nginx Menggunakan Apache Jmeter. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 8(2), 87–92.
- Saputra, A. (2012). Manajemen basis data mysql pada situs ftp lapan bandung. *Berita Dirgantara*, 13(4).
- Sha, F., Yu, K., Zhang, L., & Wu, X. (2010). A performance evaluation method and it's implementation for web service. *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, 218–222.