Jurnal TAM *(Technology Acceptance Model)*
Jurnal TAM, Volume 16, Number 1, July 2025
E ISSN: 2579-4221; P ISSN: 2339-1103, pp. 1-9
Accredited SINTA 4 Number 225/E/KPT/2022
https://jurnal.ftikomibn.ac.id/index.php/JurnalTam/index

TESTING OF PIJAR SEKOLAH APPLICATION WITH LOAD TESTING METHOD USING LOCUST

Rahmat Prastyo, Kusrini, Kusnawi
[1,2.3]Teknik Informatika, Universitas Amikom Yogyakarta
[1,2,3]Jl. Ring Road Utara, Ngringin, Condongcatur, Depok, Sleman, Daerah Istimewa Yogyakarta
E-mail: rahmatprastyo@students.amikom.ac.id, kusrini@amikom.ac.id, khusnawi@amikom.ac.id

Article history:
Received: January 22, 2025
Revised: March 18, 2025
Accepted: April 9, 2025

Corresponding authors
rahmatprastyo@students.amikom.ac.id

Keywords:
Pijar Sekolah;
load testing;
locust;
application performance.

Abstract
The Pijar Sekolah application is an application created to help the learning and teaching process. Pijar Sekolah has several features, such as attendance, assignments, exams, and grades. The feature that is widely accessed by users is the exam feature. Therefore, when many users use the application simultaneously, it is important to conduct performance test on the Pijar Sekolah application. This study purpose is to conduct performance test with the Load Testing method using Locust. Test was carried out with a gradually increasing number of users, the number of users as testers were 50, 100, 200, 400, and 800 with a ramp-up period of 1 second. The testing will be carried out in accordance with the examination process carried out using the Pijar Sekolah application by accessing Login, Login Status, Exam List, Start Exam, Question List, Exam Questions, and Submit Answers. The results of the test show that the performance in terms of response time is stable when testing from 50 to 400, but in RPS (Request Per Second) the average value increases with the number of 800 users getting an average value of 33.83 RPS. However, the test with the number of 400 users get an error in submitting answers. When test with the number of users 800, the response time increases and there are several errors by getting responses of 502 and 422 for 0.33%. The results of this study can be used to determine which processes need to be improved in performance. So that the Pijar Sekolah application can be used by many schools in carrying out the exam process simultaneously.

1. INTRODUCTION

The Pijar Sekolah application has been developed since 2019 until now. A Pijar Sekolah application aims to digitize the learning and teaching process also school management. Pijar Sekolah is a digital application launched by PT. Telkom to realize an increase in the quality of learning in schools. As an educational platform, Pijar Sekolah has many features [1]. In 2024, the number of schools registered in the Pijar Sekolah application was 2416 schools and 1,477,136 users were students, 43,928 were teacher users. In one school, there were a maximum of 721 students. The more schools that register to use Pijar Sekolah, the more users like teachers, students and school management in the following year.

In the Pijar Sekolah application, there are several modules, including teaching and learning contents, teacher and student attendances, learning assignments, assessments, and exam modules. The module that is most accessed simultaneously in Pijar Sekolah is the exam module. This online exam system model can be interpreted as an effort to create a new system to replace the old system with the overall target or improvement of the existing system so it runs well [2]. Because more and more schools are using Pijar Sekolah in conducting exams, the system needs to be tested on the module. The performance testing that will be carried out is the service on the Pijar Sekolah student exam API.

API and Web Service acts as a component that connects the interface with the server used to support procedures and functions that will process requests and responses. API (Application Programming Interface) is one of the factors that affects the performance of an application system [3].

Testing an application system is important to do. Testing an application is to find out the vulnerabilities and errors of an application system [4]. Vulnerabilities and errors in the application when accessed simultaneously by a large number of users can be tested by performance test using the Load Testing method.

Load Testing is a performance test method where the system response is measured in various conditions [4]. With Load Testing, it can determine the condition of the system when there are a number of users accessing an application system simultaneously [5]. The target users in this study is whether the Pijar Sekolah can handle a target of 800 Student users who accessed the exam system.

Locust is an open-source performance testing tool. Locust is made using the Python programming language [6]. The way locust works is very easy to use and simple in defining test scenarios. Because it uses Python it is possible to use many different protocols.

Based on the background of this study, the Pijar Sekolah API service is tested on the exam module. The tests are carried out by simulating one school conducting the exam phase. The simulation scenario to test the performance of the Pijar Sekolah application will use Locust. The purpose of this study is to test and ensure that the ongoing exam process in one school which has obstacles or does not get it when students take the exam using the Pijar Sekolah application. An analysis of the results from the application performance test will be carried out using the Load Testing method. The result can be a references to develop the Pijar Sekolah application to prepare resources and make sure the server can operate under high workload such as surge of users or requests.

II. LITERATURE

2.1. Application Programming Interface

Application Programming Interfaces, or APIs, are at the core of this progression since they enable various apps to communicate with one another and, therefore, enabling technology companies to provide a variety of practical and easy services to their clients.

The implementation of web services with REST architecture allows the development of simple, scalable, effective, safe, and reliable systems so that the design of simple and robust APIs based on Ajax and Restful web services becomes easier. The REST architecture allows the system that makes the request to access and manipulate the representation of the web resource using HTTP methods such as GET, POST, PUT, and DELETE [8].

2.2. Performance Testing

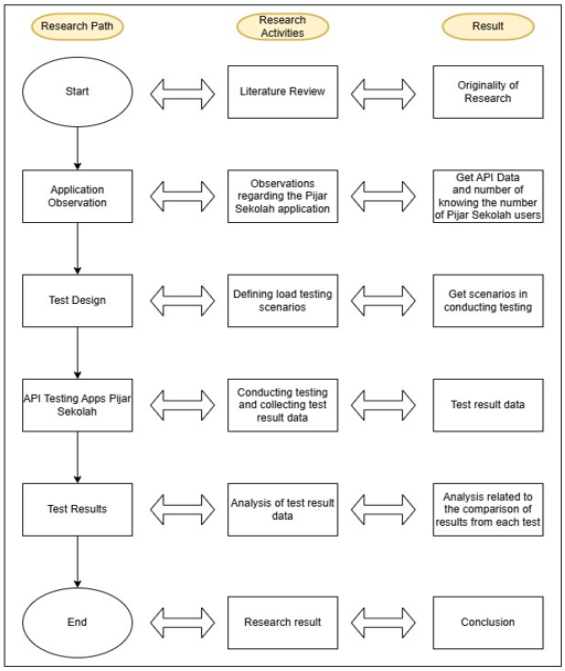
Performance testing can be done to find out the response time and throughput of each application. Performance testing tools are used to determine the time required to perform a task by a system that is usually created from an API Service to access the server. Performance testing is to check whether the system meets the non-functional requirements identified in the Software Requirements Specification (SRS) document or note. For a website, performance testing can check its speed, reliability, and load handling capacity. There are various types of performance testing: Stress testing, load testing, force testing, and volume testing. Testing tools allow testers to create, manage, and run tests for a specific environment, which is managed for specific testing for a specific application [9].

2.3. Locust

Locust is the best open-source tool for our purposes due to its outstanding performance. Software development can be done with a mix of work from Locust documentation and instructions provided by partner companies. The most important aspect of Locust is the construction of the load form and the metrics to be recorded [10].

III. RESEARCH METHODS

At the stage of the research method that will be carried out as seen in Picture 1. The research method consists the stages of research flow, stages of research activities and also the results of each research activity. The stages of this research include: observations related to the application of school glow, designing test scenarios, conducting tests, and analyzing data results from Load Testing.



Picture 1. Research Flow

The first stage of this research is to conduct a literature study. Literature study is a research activity consists of similar research studies that have been conducted by previous researchers. The literature study taken is related to application testing using API service using the Load Testing method.

The second stage is to conduct observations on the Pijar Sekolah application. The observation purpose is to obtain information related to users of the Pijar Sekolah application and how the Pijar Sekolah application system works. The data obtained from the observation will be used as a basis for conducting research by designing a Load Testing test on the Pijar Sekolah application.

Load Testing is done by using Locust (Locust.io). The test scenario will be created by using the python programming language and will be run using locust.io. Testing in this study will be carried out according to the scenario in table 1. In addition, the scenario will set variations in the 1-second ramp up period, which is the time required to add a new user to the test, and the loop count, which determines how many times each user will run the test scenario [6].

Table 1. Test Scenario Table

API Request	Scenario 1	Scenario 1	Scenario 1	Scenario 1	Scenario 1
Login	50 user	100 user	200 user	400 user	800 user
Status Login	50 user	100 user	200 user	400 user	800 user
Exam List	50 user	100 user	200 user	400 user	800 user
Mulai Ujian	50 user	100 user	200 user	400 user	800 user
Exam Question List	50 user	100 user	200 user	400 user	800 user
Exam Question	50 user	100 user	200 user	400 user	800 user
Submit Answer	50 user	100 user	200 user	400 user	800 user

The final stage of this research is the analysis of the results from the load testing findings from each scenario and can be used as evaluation material for the development of the exam module from the Pijar Sekolah application.

IV. RESULTS

In this research, the load testing method was used using Locust. The script that has been created using Locust will be executed which will be used to measure the performance of the Pijar Sekolah application by sending HTTP requests using the API service in a large number of Concurrent Users.

According to each API, a test scenario is created using python. The test scenario is created starting from the Login API, Login Status API, Exam List API, Start Exam API, Question List API, Exam Question API, and the last is the Submit Answer API.

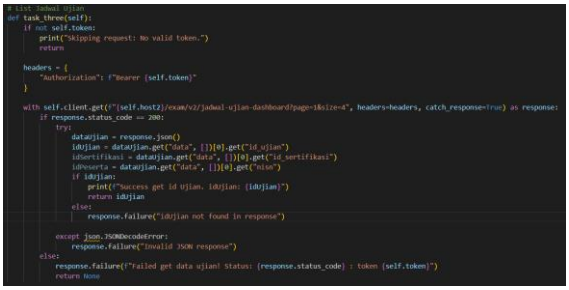
The first is the completion of the test script for the login API in picture 2, which functions to obtain an access token to access other APIs.

Picture 2. Test Script of Student Login

After the login API has been successfully created, the next step is the login status API in Figure 4, which has the function of obtaining information on successful login and gaining access.

Picture 3. Test Script of Status Login

After the API login status is successfully accessed (Picture 3), you can proceed to the next stage, which is the API List Exam Schedule script that has been created (Picture 4). The API List Exam Schedule is used to obtain information in the form of an exam ID.

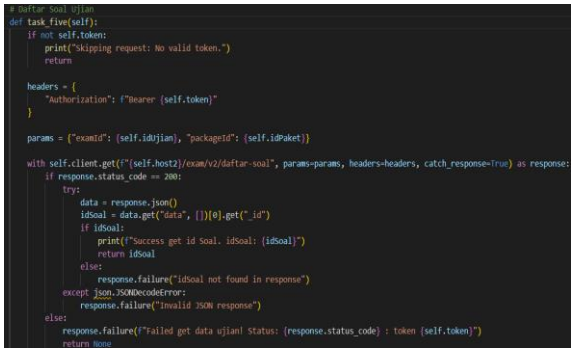


Picture 4. Test Script of Exam Schedule Lists

After successfully accessing the Exam Schedule API List and getting the Exam ID, the next step is to access the Start Exam API based on the exam ID has been obtained.

Picture 5. Test Script of Start Exam

The Start Exam API created in Picture 5 functions to obtain information on the question package ID contained in the exam ID. After successfully obtaining the package ID, the next step is to access the Question List API according to the package ID that has been obtained in the Start Exam API response. The question list script has been completed in Picture 6.



Picture 6. Test Script of Exam Questions List

After the Question List API is successfully accessed, it will display information in the form of the ID of each question item to Picture 7.

Picture 7. Test Script of Exam Questions

After getting the question ID information, the next last step is to submit the exam answer. This API will submit the question answer according to the question ID, package ID, and exam ID that have been obtained from each API that has been successfully accessed. The script to submit the answer is according to Picture 8.

Picture 8. Test Script of Answer Submit

After the scenario for each API request is successfully created with the Python programming language, the script will then be executed using Locust. The results of the execution of each test scenario that has been created are as follows:

After creating a test script for several APIs that will be used in testing the Pijar Sekolah application in the exam module, it will then be executed to obtain the results of each scenario that has been created in the Pijar Sekolah exam API testing plan.

Locust is used to measure application performance by using API requests to send HTTP requests to a large number of users simultaneously. Performance of the application's response time to users when performing the test process. And errors obtained when performing the test process. Testing is done by entering the number of users gradually from 50, 100, 200, 400, and 800 users.

1. Test Results with 50 Users in performing the test

The test results with locust on 50 users performing the exam process by accessing Login, Login Status, Exam List, Start Exam, Exam Question List, Exam Questions, and Submit Answers. In Table 2, there are no errors that occur in each request.

Furthermore, the test results show that each API request has a different response time - the fast API response is the Login Status API with an average response time of 13.58 ms, then the slowest average response is in the Login API with 33.33 ms. And other API responses include the List API Start Exam List with an average response of 28.35 ms, Start Exam API with an average response of 19.28 ms, Exam Question List API 19.48 ms, Exam Question API 17.04, and the last is the Submit Answer API with an average response of 26.27.

RPS (Requests Per Second) for 50 users with an average of 2. And the longest request is in the Login API, which is 3.2 requests per second.

Table 2. Test Result from 50 Users

API	Fails	Median	95%-ile (ms)	99%-ile (ms)
Login	0	23	72	120
Status Login	0	13	19	35

Exam List	0	19	62	95
Mulai Ujian	0	17	24.7	24.8
Exam Question List	0	18	24.22	24.22
Exam Question	0	16.0821	18.66	18.67
Submit Answer	0	26	33	43
Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS
33.33	20.04	119.52	1207	3.2
13.58	9.26	32.81	1471.1	2.3
28.35	14.28	94.72	774	3.1
19.28	12.26	41.97	293	2.9
19.48	14.08	45.43	596	2.9
17.04	12.69	32.83	1012.9	1.9
26.27	20.26	42.89	90	1.7

2. Test Results with 100 Users performing the test

The number of users is increased to get the test results with locust at the number of 200 users doing the test process by accessing Login, Login Status, Exam List, Start Exam, Exam Question List, Exam Questions, and Submit Answers. In table 3 on each request there is no error that occurs.

Furthermore, the results of testing with 100 users show that the fast API is still in the Login Status API with an average response time of 14.06 ms, then the longest average response is in the Login API with 30.32 ms. And other API responses include the Exam List API with an average response of 25.37 ms, the Start Exam API with an average response of 18.22 ms, the Exam Question List API 18.75 ms, the Exam Question API 16.84 ms, and the last is the Submit Answer API with an average response of 25.53 ms.

With the increasing number of RPS (Requests Per Second) for the number of users 100 increased by an average of 4.5. And the longest request is in the Login API which is 5.2 requests per second.

Table 3. Test Result from 100 Users

API	Fails	Median	95%-ile (ms)	99%-ile (ms)
Login	0	23	69	110
Status Login	0	13	20	100
Exam List	0	18	58	70
Mulai Ujian	0	16	24	24
Exam Question List	0	17	24	24
Exam Question	0	16	20	20
Submit Answer	0	25	33	49
Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS
30.32	19.39	111.17	1207	5.2
14.06	9.43	100.06	1474	4.2
25.37	13.60	134.82	774	5
18.22	12.55	88.47	293	4.9
18.75	12.29	63.39	596	4.8
16.84	11.72	42.51	1018	3.8
25.53	19.39	49.21	90	3.6

3. Test Results with 200 Users performing the test

The test results with locust on 200 users conducting the exam process by accessing Login, Login Status, Exam List, Start Exam, Exam Question List, Exam Questions, and Submit Answers on. In Table 4, there are no errors that occur in each request.

Furthermore, the results of a test with 100 users show that the fastest API is still in the Login Status API with an average response time of 14.37 ms, then the slowest average response is in the Login API with 28.48 ms. And other API responses include the Exam List API with an average response of 22.99 ms, the Start Exam API with an average response of 18.24 ms, the Exam Question List API 18.92 ms, the Exam Question API 17.22 ms, and the last is the Submit Answer API with an average response of 25.35 ms.

By the increasing number of RPS (Requests Per Second) for the number of users 200 increased by an average of 7.78. And the longest request is in the Login API which is 8.7 requests per second.

Table 4. Test Result from 200 Users

API	Fails	Median	95%-ile (ms)	99%-ile (ms)
Login	0	23	62	79
Status Login	0	13	22	30
Exam List	0	18	54	70
Mulai Ujian	0	16	26	26
Exam Question List	0	17	26	26
Exam Question	0	16	24	24
Submit Answer	0	24	33	46
Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS
28.48	18.17	129.53	1207	8.7
14.37	9.52	337.83	1475	7.5
22.99	126.607	207.41	774	8.3
18.24	11.66	108.41	293	8.2
18.92	12.49	91.13	596	8.09
17.22	11.03	87.79	1019	7
25.35	18.29	95.46	90	6.8

4. Test Results with 400 Users performing the test

The test results with locust on 400 users who accessed the exam process by accessing Login, Login Status, Exam List, Start Exam, Exam Question List, Exam Questions, and Submit Answers. In Table 5, the Submit Answer API request had an error of 0.002443%.

Furthermore, according to the results of the average response time for testing with 400 users, the fast API in the Exam Questions API has an average response time of 19.93 ms, then the longest average response is in the Login Status API with 29.22 ms. And other API responses include the Login API with an average response of 29.11 ms, the Exam List API

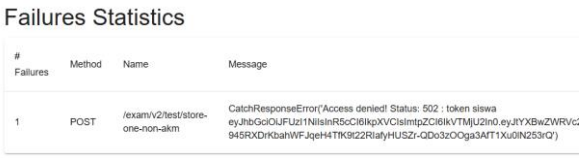
with an average response of 23.72 ms, the Start Exam API with an average response of 21.7 ms, the Exam Questions List API 21.96 ms, and the last is the Submit Answer API with an average response of 29.11 ms.

Table 5. Test Result from 400 Users

API	Fails	Median	95%-ile (ms)	99%-ile (ms)
Login	0	25	40	98
Status Login	0	13	59	82
Exam List	0	18	52	92
Mulai Ujian	0	16	44	56
Exam Question List	0	18	45	65
Exam Question	0	16	38	50
Submit Answer	0,002443	25	40	98
Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS
29.11	18.14	954.29	1207	15.8
29.22	9.08	618.14	1475	14.7
23.72	12.64	693.18	774	15.5
21.7	11.41	819.46	293	15.4
21.96	11.994	752.03	596	15.8
19.93	10.47	635.46	1011	14.5
29.11	18.14	915.83	90	14

With the increasing number of RPS (Requests Per Second) for the number of users 400 increased by an average of 15.1. And the longest request is in the Login API and the Exam Question API which is 15.8 requests per second.

The error from the request that occurred on the Submit Answer API was getting a 502 bad gateway response in picture 9.



Picture 9. Error Response from 400 users

5. Test Results with 800 Users performing the test

In Table 6, the test results with Locust on 800 users conducting the exam process by accessing Login, Login Status, Exam List, Start Exam, Exam Question List, Exam Questions, and Submit Answers. In each API request accessed by 800 users, there are several APIs that encounter error constraints like the Login API of 0.00125%, the Exam List API of 0.01%, the Exam Question List API of 0.010012%, the Exam Question API of 0.008053%, and also the Submit Answer API of 0.00125%. There are 2 APIs that are still safe when there are 800 users conducting the exam process from Login, namely the Login status API and the Start Exam API. Several error responses can be seen in the table. 6 Error Request 800 users.

Table 6. Test Result from 800 Users

API	Fails	Median	95%-ile (ms)	99%-ile (ms)
Login	0,00125	59	250	390
Status Login	0	30	130	200
Exam List	0,01	42	220	440
Mulai Ujian	0,0025	38	143	143
Exam Question List	0,010012	30	143	143
Exam Question	0,008053	28	134	134
Submit Answer	0,00125	42	200	670
Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS
94.73	19.31	775.71	1206	40.06
46.24	9.17	552.48	1468	31.35
80.26	12.42	1696.55	742	37.93
66.65	11.73	1620.03	292	38.88
68.49	12.43	1606.97	593	35.72
71.32	11.12	1396.27	1091	26.98
75.37	18.46	1553.95	90	25.92

Data from the results of the average response time for a test using 800 users, the fast API in the Login Status API has an average response time of 46.24 ms, then the slowest average response is in the Login API with 94.73 ms.

By the increase the number of RPS (Requests Per Second) for the number of users 800 increased by an average of 33.83 and the longest request is in the Login API and the Exam Question API which is 40.06 requests per second.

The errors obtained in Table 7 are 26 requests with 1 Login API request, 8 Exam List API requests, 2 Start Exam API requests, 7 Question List API requests, 7 Exam Question API requests, and 1 Submit Answer API request. The total error when there are 800 students taking the exam is 0.033%. And other API responses include the Exam List API with an average response of 80.26 ms, the Start Exam API with an average response of 66.65 ms, the Exam Question List API 66.49 ms, the Exam Question API 71.32 ms, and the last is the Submit Answer API with an average response of 75.37 ms.

Table 7. Error Testing List from 800 Users

Method	Name	Error	Occurrences
GET	/exam/v2/test/getSoal?idUjian=6788d19762d3bf001ffd2b83&idPaket=6788d14d62d3bf001ffd27c2&idSoal=6788d15e62d3bf001ffd27df	CatchResponseError('Access denied! Status: 502')	1
GET	/exam/v2/test/getSoal?idUjian=6788d19862d3bf001ffd2dba&idPaket=6788d14d62d3bf001ffd27c2&i	CatchResponseError('Access denied! Status: 502')	1

Method	Name	Error	Occurrences
	dSoal=6788d15e62d3bf001ffd27df		
GET	/exam/v2/test/getSoal?idUjian=6788d19862d3bf001ffd2e10&idPaket=6788d14d62d3bf001ffd27c2&idSoal=6788d15e62d3bf001ffd27df	CatchResponseError('Access denied! Status: 502')	1
GET	/exam/v2/daftar-soal?examId=6788d19862d3bf001ffd2e69&packageId=6788d14d62d3bf001ffd27c2	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/test/getSoal?idUjian=6788d19862d3bf001ffd2d49&idPaket=6788d14d62d3bf001ffd27c2&idSoal=6788d15e62d3bf001ffd27df	CatchResponseError('Access denied! Status: 502')	1
GET	/exam/v2/test/start-ujian/6788d19862d3bf001ffd2e6c	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/daftar-soal?examId=6788d19762d3bf001ffd2b96&packageId=6788d14d62d3bf001ffd27c2	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1
POST	/exam/v2/test/store-one-non-akm	CatchResponseError('Access denied! Status: 502')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/test/start-ujian/6788d19862d3bf001ffd2e66	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/daftar-soal?examId=6788d19762d3bf001ffd2ca8&packageId=6788d14d62d3bf001ffd27c2	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/test/getSoal?idUjian=6788d19762d3bf001ffd2c62&idPaket=6788d14d62d3bf001ffd27c2&idSoal=6788d15e62d3bf001ffd27df	CatchResponseError('Access denied! Status: 502')	1
POST	/student/login	CatchResponseError('Login failed! Status: 502 : username 221101142 : password 221101142')	1
GET	/exam/v2/daftar-soal?examId=6788d19862d3bf001ffd2e6c	CatchResponseError('Failed get data ujian! Status: 422')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/daftar-soal?examId=6788d19862d3bf001ffd2e66	CatchResponseError('Failed get data ujian! Status: 422')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/daftar-soal?examId=6788d19762d3bf001ffd2c1f&packageId=6788d14d62d3bf001ffd27c2	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/daftar-	CatchResponseErr	1

Method	Name	Error	Occurrences
	soal?examId=6788d19862d3bf001ffd2e21&packageId=6788d14d62d3bf001ffd27c2	or('Failed get data ujian! Status: 502')	
GET	/exam/v2/daftar-soal?examId=6788d19762d3bf001ffd2ca7&packageId=6788d14d62d3bf001ffd27c2	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1
GET	/exam/v2/test/getSoal?idUjian=6788d19762d3bf001ffd2c27&idPaket=6788d14d62d3bf001ffd27c2&idSoal=6788d15e62d3bf001ffd27df	CatchResponseError('Access denied! Status: 502')	1
GET	/exam/v2/jadwal-ujian-dashboard?page=1&size=4	CatchResponseError('Failed get data ujian! Status: 502')	1

V. CONCLUSION

Based on the results of the load test, the resilience and reliability of the Pijar Sekolah application in carrying out the exam process by using the application is good enough to handle user loads from 50 to 400. Although the result responses vary, it increases with the increase in requests on each Pijar Sekolah API. The fastest response when accessing the Login Status API and the slowest response when accessing the Login API. And in the last scenario test, where 800 users took the exam process, several users experienced errors by getting responses of 504 and 422 with a total of 0.033% of users having problems in the process of taking the exam using the Pijar Sekolah application. To handle the obtained errors, improvements need to be made to the process of this application. So that further research can be carried out which can be focused on reducing variations in response time and minimizing errors that occur when there are more than 800 users taking the exam using the Pijar Sekolah application. Conduct regular load tests to monitor the application's performance, especially as the number of users increases. Use monitoring tools to detect potential issues in real-time.

REFERENCES

[1] Liesye A., & Waskito., 2024. Inovasi Pemakaian Aplikasi Pijar Dari PT. Telkomel Untuk Pelaksanaan Evaluasi Pembelajaran Secara Online di SMKN 1 Kecamatan Luak. Journal Of Socual Science Research. 4(1), pp. 9794-9803.

[2] Tommy I P., 2022. Penggunaan Aplikasi Sistem Seleksi Elektronik (SSE) Pada Pelaksanaan Seleksi Masuk Jalur UM-PTKIN Tahun 2022. EDU-MANDARA, 1(2), pp. 117-122.

[3] Arlinta C B., Johannes H., Efren M., 2021. PENGUJIAN API WEBSITE UNTUK PERBAIKAN PERFORMANSI APLIKASI DITENUN. Journal of Applied Technology and Informatucs, 1(3), pp.14-21.

[4] D. I. Permatasari et al., "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian," J. Sist. dan Teknol. Inf., vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.

[5] Setiawan, G.H., I Made, B.A., & Budiarta K., 2022. Pengujian Performa API (Application Programming Interface) dengan Metode Load Testing. Seminar Nasional Corisindo. Bali.

[6] Jonas E., 2023. An evaluation of tools for verifying non-functional requirements for cloud deployed applications. University UMEA.

[7] Sabella B., Tina. R.R., Achmad F., Lestari A., & Fathurrahmani., 2024. Pengujian Aplikasi SIHARAPAN Menggunakan Metode Stress Testing. Jurnal Teknologi Informasi dan Terapan (J-TIT) Vol. 11 No. 1 Juni 2024 ISSN: 2580-2291.

[8] Neumann, A., Laranjeiro, N. and Bernardino, J., 2018. An analysis of public REST web service APIs.IEEE Transactions on Services Computing,14(4), pp.957-970.DOI: https://doi.org/10.1109/TSC.2018.2847344.

[9] Tanuj Wala., 2014. "A Comparative Study of Web Service Testing Tools", International Journal of Advanced Research in Computer Science and Software Engineering 4(2).

[10] Sebastian F., 2023. Load testing on an alarm server. Computer Science and Engineering. University Halmtad.

[11] Maharani, C, N., & Darwis D., 2023. Analisa Perbandingan Kualitas Perangkat Lunak Pada Website Perguruan Tinggi Menggunakan Metode Webqual, Apache J-Meter, dan Web Server Stress Tool. Jurnal Teknologi Dan system Informasi. DOI: https://doi.org/10.33365/jtsi.v4i1.2436

[12] Kamarudin., Kusrini., & Sunyoto A., 2018. Uji Kinerja Sistem Web Service Pembayaran Mahasiswa Menggunakan Apache JMeter (Studi Kasus: Universitas Amikom Yogyakarta). Jurnal Teknologi Informasi. ISSN: 1907-2430.

[13] Amien, J, A., Winarso D., 2019. ANALISA PENINGKATAN KINERJA FTP SERVER MENGGUNAKAN LOAD BALANCING PADA CONTAINER. Jurnal Fasilkom Vol.9 No. 3 November 2019 ISSN: 2089-3353.

[14] Madhani D., Darwinyanto E., & Gandhi A., 2023. Performance Testing Menggunakan

Metode Load Testing dan Stress Testing pada Sistem Core Banking PT. XYZ. e-Proceeding of Engineering, 10(6), pp. 5431-5441.

[15] Yenugula M., Kodam R., & He D., 2019. Performance and Load Testing : Tools and Challenges. Internasional Journal of Engineering in Computer Science 2019, 1(1), pp. 57-62.

[16] Hendayun M., Ginanjar A., & Ihsan Y. 2023. ANALYSIS OF APPLICATION PERFORMANCE TESTING USING LOAD TESTING AND STRESS TESTING METHODS IN API SERVICE. Jurnal Sisfotek Global, 13(1), pp. 28-34. DOI: http://dx.doi.org/10.38101/sisfotek.v13i1.2656.

[17] Ismail A., Ananda, Y, A., Arieh, S, N., & Hamdana, E, N., 2023. PERFORMANCE TESTING SISTEM UJIAN ONLINE MENGGUNAKAN JMETER PADA LINGKUNGAN VIRTUAL. Jurnal Informa Polinema, 9(2), PP. 159-164.

[18] Santos R., Brown, M, A., 2017. Oracle® Load Testing Load Testing User's Guide Release 13.2.0.1. America: Oracle.