Bioinformatics Database Query Performance and Optimization

Edy Budiman, Andi Tejawati, Ummul Hairah

Abstract*: Borneo bioinformatics portal test is a critical element of SQA and represents a comprehensive review of specifications, design and coding. The test represents an abnormality in the development of the portal. A series of tests systematically reveals several different types of errors. This study aims to evaluate the performance and optimization of Borneo's Bioinformatics portal with a series test activities using the Web Performance Optimization methodology. Testing query performance with measuring the response time and page loading timings from the object relationship mapping (ORM) model Laravel PHP framework in offline and online. For optimization, we set a pre-test and post-test scenario to evaluate the efficiency performance test results. The results study found that the query relation model, parsing script (javaScript and CSS), service scale and dimension images in the interaction process to the database are the dominant resources affecting the performance of the Bioinformatics portal. Performance optimization through determining the appropriate query relation model, minify and defer parsing script or combine images using CSS sprites to reduce scala image.

Keywords* : Bioinformatics, query, database relationship, ORM.

I. INTRODUCTION

One of the optimization parameters to improve the performance of a bioinformatics resource portal is the database query model. In In overcoming the challenges of complex queries, several methods of query RDMBS modelling proposed in [1], under bioinformatics domains.

Various approaches can be applied to improve the performance of bioinformatics portals, can uses perform custom queries with a compilation of bioinformatics scenarios[2], database integrated system based on SOAP web service[3] and extensive web services[4], resource portal[5], temporal database[6] or via caching[7], DOM interactive and loaded time content[8], the graph-based database partitioning method (parallel query)[9], efficiency of the Biodiversity resource portal with of the key performance indicators[10], query optimization through the Grid data sources architecture[11], taxonomy of plants Nomenclature[12], NoSQL Biological Databases[13], with ontology and taxonomy model[14] and Ontologies for Query Relaxation[15], etc.

Revised Manuscript Received on September 25, 2020.

***** Correspondence Author

Edy Budiman*, Informatics Department, Universitas Mulawarman, Samarinda, Indonesia. Email: edy.budiman@fkti.unmul.ac.id

Andi Tejawati, Informatics Department, Universitas Mulawarman, Samarinda, Indonesia. Email: andi.tejawati@gmail.com

Ummul Hairah, Informatics Department, Universitas Mulawarman, Samarinda, Indonesia. Email: ummihairah@gmail.com

The query optimization method we propose is a file resource efficiency issue or source-code used in the development of bioinformatics webportal.

The bioinformatics portal testing objects was portal was developed by the author to manage data on plant biodiversity on the Borneo island[16]. The portal is built using the Model-View-Controller (MVC) concept based on Laravel's PHP framework and utilizes the Eloquent ORM features in query relations data mapping[17]. Now, the Borneo's Bioinformatics Portal currently contains 233 record medicinal plant, 1482 record tree species and 86 record wood species include 80 record bamboo species[18].

This study aims to evaluate with performance testing of the query model and optimizes the script (source code) on the Borneo Bioinformatics portal. For query performance testing we measure the response time of the query relations the Laravel eloquent ORM model and page load Timings of Borneo Bioinformatics portal. Whereas for script optimization, we measure the efficiency of source code usage; PHP, JavaScript and CSS using the pretest and posttest scenarios in Borneo's Bioinformatics portal.

Research contribution: A common problem of the Bioinformatics portal is web performance optimization. This study result is a proposal for a query optimization method on the Bioinformatics web portal which is built within the PHP Laravel framework, through an efficiency optimization approach to source-code (scripts) or other data resource files.

II. RELATED WORK

A. Literature

A brief literature review of related work in the area of Bioinformatics portal and database performance evaluation resulted in the identification of several studies, such as; Bultet, L. et al, the SIB Swiss Institute of Bioinformatics' resources: focus on curated databases[19], Bioinformatics web portals[20], ExPASy: SIB bioinformatics resource portal[21], Ethnobotany database: Exploring diversity medicinal plants of Dayak Tribe Borneo[22], Manycore High-Performance Computing in Bioinformatics[23], A comparison of microbial genome web portals[24], BioPortal: A Portal for Deployment of Bioinformatics Applications on Cluster and Grid Environments[25], mobile internet services performance in borneo[26], etc. Other than that Sukhpuneet Kaur et. al An Empirical Performance Evaluation of Universities Website International Journal of Computer Applications[27], and D.

Luna in Government web portals performance evaluation using data envelopment analysis[28], where according to these studies the quality assurance of a website depends on automated testing tools which lower costs and increase its efficiency. The performance of a website can be an important factor for its success. It depends on the main speed factor. If the website speed is fast then the performance automatically increases.

B. Metric for Perfomance

Performance can be evaluated using tools that break down the resources and components on the bioinformatics web site. There are a wide variety of automated site testing tools available.

- *Google PageSpeed Insights (PSI),* Refers to [29] PageSpeed Insights (PSI) reports on the performance of a page on both mobile and desktop devices and provides suggestions on how that page may be improved. PSI provides a score which summarizes the page's performance. This score is determined by running Lighthouse to collect and analyze lab data about the page. A score of 90 or above is considered good. 50 to 90 is a score that needs improvement, and below 50 is considered poor. PSI also classifies field data into 3 buckets, describing experiences deemed good, needs improvement, or poor (see Tabel-I).

Tabel- I: Classifying Good, Needs Improvement, Poor[29]

	Good	Needs Improvement	Poor
FCP	[0, 1000ms]	(1000ms, 3000ms]	over 3000ms
FID	[0, 100ms]	(100ms, 300ms]	over 300ms
LCP	[0, 2500ms]	(2500ms, 4000ms]	over 4000ms
CLS	[0, 0.1]	(0.1, 0.25]	over 0.25

- *Yslow from Yahoo,* Refers to [30] YSlow grades web page based on one of three predefined rulesets or a user-defined ruleset. It offers suggestions for improving the page's performance, summarizes the page's components, displays statistics about the page, and provides tools for performance analysis [31]. YSlow's web page analysis is based on the 23 of these 34 rules that are testable [30] (see Table-II).

Tabel -II. Rule weights of YSlow V2 Ruleset [32].

Rule	Compress component with GZip	Avoid CSS expressions	Minify JScript and CSS	Remove duplicate JScript and CSS
(A) 90 <= S <= 100	0 file size < 500b	0 to 5 expressions on CSS or inline STYLE	0 or 1 unminified component	0 to 2 duplicated JS or CSS
(B) 80 <= S < 90	1 file size < 500b of any type	6 to 10 expressions on CSS or inline Style	2 unminified components	3 or 4 duplicated JS or CSS
(C) 70 <= S < 80	2 file size < 500b	11 to 15 expressions on CSS or inline Style	3 unminified components	5 or 6 duplicated JS or CSS
(D) 60 <= S < 70	3 file size < 500b of any type	16 to 20 CSS or inline Style	4 unminified components	7 or 8 duplicated JS or CSS
(E) 50 <= S < 60	4 uncompressed or file size < 500b	21 to 25 expressions on CSS or inline Style	5 unminified components	9 or 10 duplicated JS or CSS
(F) 0 <= S < 50	>= 5 file size < 500b	>=26 CSS or inline Style	>= 6 unminified components	>= 11 duplicated JS or CSS

Matrix table-keys according to[32]:

- Rule: The YSlow performance rule
- Weight. How this performance rule is weighted in the overall page analyis grade
- Points. Number of points deducted per offender (performance infraction occurance), from a total of 100 per rule
- Score Computation. The formula used to compute the final score per rule
- Grades from A to F. How many components/offenders is necessary to reach grades from A to F.

C. Borneo Bioinformatics Portal

One of the efforts to manage biodiversity is through data and information support. Data and information on biodiversity need to be continued efforts are made to be added, both in species diversity, habitat, population, and distribution. Records of 47,910 species of Indonesian biodiversity [33] are estimated still far less than the potential that actually exists. It is necessary to increase the intensity of the implementation of identification and inventory of biodiversity in the field, and on the other hand, a database system that is able to collect data and information that is spread across various circles is needed

Borneo bioinformatics portal as a data management system and information on endemic plants for the island of Borneo, Kalimantan, Indonesia. Borneo Bioinformatics is an example of an information system which presents taxonomic data, form an ontology model which can serve, mapping data as information regarding data descriptions and relationships between taxons in accordance with taxonomic levels based on data stored in the database[34].

The portal online is accessed in url https://www.borneodiversity.org/index. The BBIS interface is shown in Figure 1.

Fig. 1. Screenshot of the borneo bioinformatics portal

The first stage of data collection, to date, the system have recorded 233 Medicinal data, 1482 tree species, 86 types of wood, and 80 types of bamboo. Until now, the data collection process is still continuing.

III. METHODOLOGY

This study evaluates the performance of the query model and optimizes the script on the Borneo Bioinformatics portal. For query performance testing, we measured the response times of the query Object Relationship Mapping (ORM) model and page loads of the Borneo Bioinformatics portal. As for script optimization, we measure the efficiency of using source code; PHP, JavaScript and CSS use the pretest and posttest scenarios on the Borneo Bioinformatics portal.

A. Performance Testing Methodologies

▪ *For query bioinformatics portal*: the Borneo Bioinformatics Portal is built based on Laravel PHP framework ("active record" pattern), We test query performance utilizing the Laravel Object Relations Mapping model features available in interactions to the database.

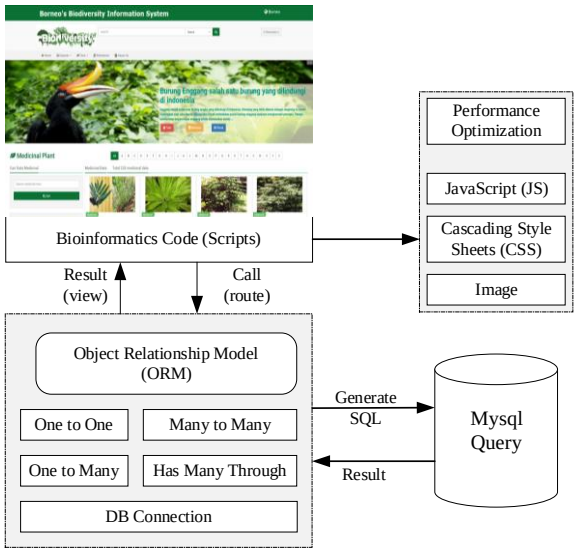

Fig. 2. Performance test design

The test is to get the query execution time (response) based on the relations formed. Two types of testing are performed, offline and online for each relationship ie; one to one, one to many, many to many, and has many through using the Laravel Debugbar package. An overview of the performance test design is seen in Figure 2.

▪ *For optimization performance bioinformatics portal*: performance testing and optimization using the pretest and post-test scenarios. Pretest is the performance of the portal before optimization, Posttest is the performance portal after optimization. Performance measurement parameters are Script(Javascript and CSS), images and page load timings, which are files that affect the performance of a Bioinformatics portal. Table 1 is the parameter as a performance test metric.

Table- III: Performance optimization metrics

Parameter	Metric
Image	Serve scaled images Optimize images Image dimensions
Script (CSS and JavaScript	Minify CSS and Javascript Defer parsing JavaScript Inline small CSS and JavaScript Combine images using CSS sprites Avoid CSS @import Duplicate JavaScript and CSS Avoid CSS expressions
Page Load	Redirect duration
Timings	Connection duration Backend duration Time to First Byte (TTFB) DOM interactive-content loaded time First paint time First contentful paint time Onload time

B. Equipment and Tools

The performance testing and optimization tools used are shown in Table IV

Table- IV: equioment and tools

Equipments	Specifications
Hardware: Laptop Platforms Server	2.5 GHz Dual Core Intel Core i5 8GB-4GB 2666MHz, 4 GB DDR4-Memory Server version: 10.1.31-MariaDB - MariaDB Serve
Software: Laravel Apache Xampp Bioinformatics portal GTMetrix Tools	 ORM Package Laravel debugbar versi 5.5 PHP version 7.1 url: http://borneodiversity.org PSI Google and Yslow yahoo

C. Performance Test Scenarios

Performance test and optimization scenarios (Query - Script) used are seen in Table V

Table- V: Performance test scenarios

Scenarios	Descriptions
Query performance: a. One-to-one b. One-to-many c. Many-to-many d. Has-Many-Through	Using Eloquent - Query Builder (ORM vs Non ORM) to data search process and view in Bioinformatics portal. Measurement data: Offline and Online a. Relation data wood to medicinal b. Relation data tree to wood c. Relation data tree to wood d. Relation data medicinal to data tree and wood

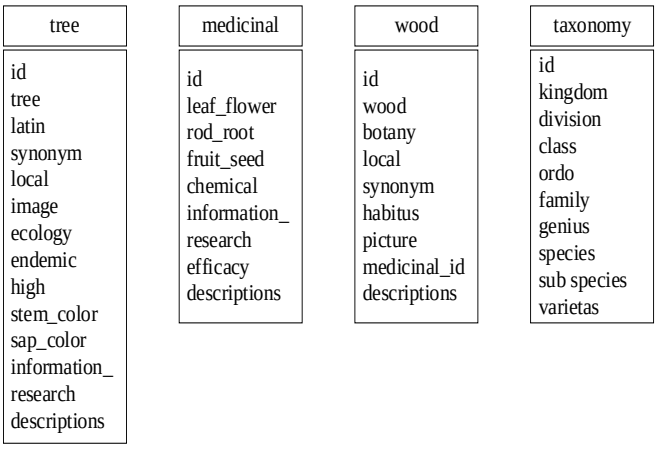

Fig. 3. Test tables for query performance (data relations)

Figure 3 for testing the relation scheme *One To One relation (hasOne),* between medicinal table and wood table, scheme One to Many relation (hasMany) between tree table and wood table, scheme Many to Many relation (belogsToMany) between tree table and wood table, and for scheme Has Many Through relation between medicinal table to tree table and wood table.

IV. RESULTS AND DISCUSSION

A. Result: Query Performance

The query test gets the execution time to display data based on the relations schema in the scenario. Tests are carried out on each relationship, ie, one to one, one to many, many to many, and has many through, and the offline and online testing.

The query performance test results are as follows:

- *Result: One-To-One relation*

This test is done to get the time from the results of the query execution for the One to One relation by using Eloquent ORM and Query Builder to display data. One-to-One relationship between wood table and medicinal table. The results of One To One relations query test are shown in Table V:

Table- V: Query performance for One To One relation

	Online		Offline	
Test	ORM	Non ORM	ORM	Non ORM
1	9710	58.63	143.31	7.25
2	6780	250.75	137.02	6.26
3	5680	324.65	148.93	6.77
4	7870	40.65	148.17	7.74
5	3550	124.81	136.12	7.27
6	4920	119.59	135.75	7.06
7	5470	140.65	137.82	7.69
8	2190	447	144.76	7.64
9	2500	574.75	137.11	7.17
10	2750	49.19	137.67	6.2
Avg.	**5142**	**213.067**	**140.666**	**7.105**

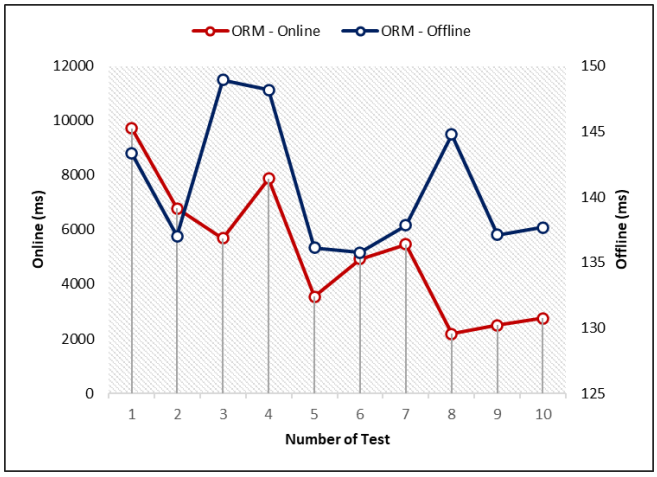

Fig. 4. Performance query ORM One to One relationship

The results of testing the ORMe One To One relationship query in Figure 4, explain the average results of the comparison performance between the use of ORM online and offline, the difference in response time average is 5595.54 ms for online, and offline of 140.66 ms query execution time, with 2 table relations showing all the data fields there is. It was found that ORM offline it is faster.

- *Result: One-To-Many relation*

This test is done to get the time from the results of the query execution for the One to Many relation by using Eloquent ORM and Query Builder to display data. One to Many relationship between tree table and wood table. The results of One To Many relations query test are shown in Table VI and Figure 5.

Table -VI: Query performance for One To Many relation

	Online		Offline	
Test	ORM	Non ORM	ORM	Non ORM
1	22930	115.6	372.59	68.02
2	20160	260.96	383.09	15.83
3	22400	328.45	383.28	66.14
4	22400	268.56	423.1	14.51
5	21420	252.57	380.3	14.84
6	33460	161.03	381.34	5.94
7	42700	367.03	387.52	6.1
8	7700	231.46	379.34	6.76
9	29420	411.13	402.37	6.58
10	25960	456.15	401.4	6.3
Avg.	**24855**	**285.294**	**389.433**	**21.102**

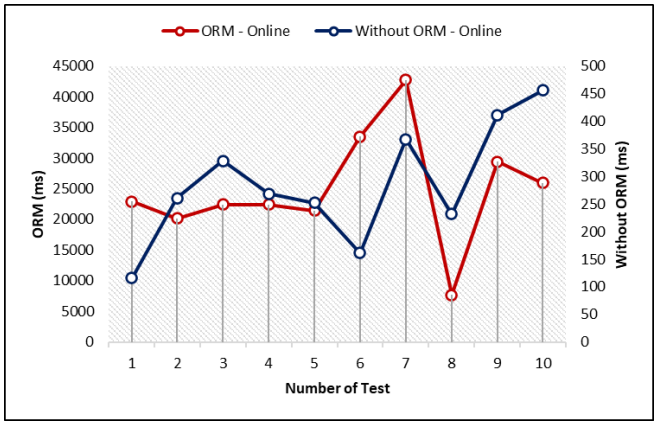

Fig. 5. Performance query One to Many relationship

From the test in Figure 5, it is known that the One To Many relation with the ORM response time of average for online is 24855 ms to execute the field data and without ORM is 285.294 ms.

- *Result: Many-To-Many relation*

This test is done to get the time from the results of the query execution for the Many to Many relation by using Eloquent ORM and Query Builder to display data. One to Many relationship between tree table and wood table. The results of Many To Many relations query test are shown in Figure 6 and Table VII.

Table- VII: Query test for Many To Many relation

	Online		Offline	
Test	ORM	Non ORM	ORM	Non ORM
1	383.74	257.24	13.77	7.61
2	740.64	157.46	14.22	5.2
3	349.78	179.4	12.45	5.47
4	411.19	259.61	12.56	4.81
5	363	288.02	11.93	4.72
6	799.72	341.32	13.27	6.79
7	228.31	318.83	12.42	5.19

8	429.79	407.85	12.21	5.16
9	146.54	232.95	11.28	6.74
10	432.81	122.81	11.91	4.97
Avg.	**428.552**	**256.549**	**12.602**	**5.667**

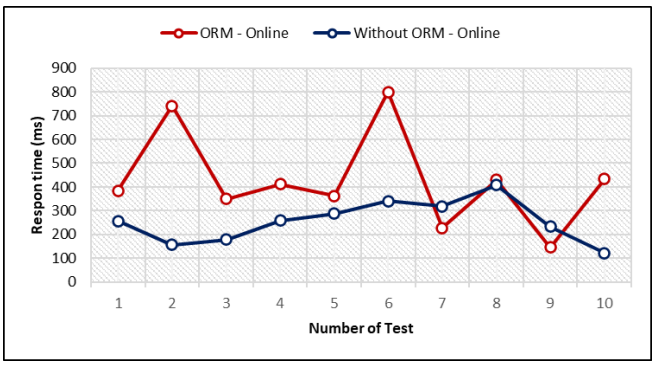

Fig. 6. Performance query Many to Many relationship

From the test in Figure 6 and Table VII, it is known that the Many To Many relation without ORM response time offline average is 5.667 ms and online average time is 256.549 ms to execute the field data.

- *Result: Has Many Through relation*

This test is done to get the time from the results of the query execution for the Has Many Through relation by using Eloquent ORM and Query Builder to display data. One to Many relationship between tree table and wood table. The results of Has Many Through relations query test are shown in Table VIII and Figure 7.

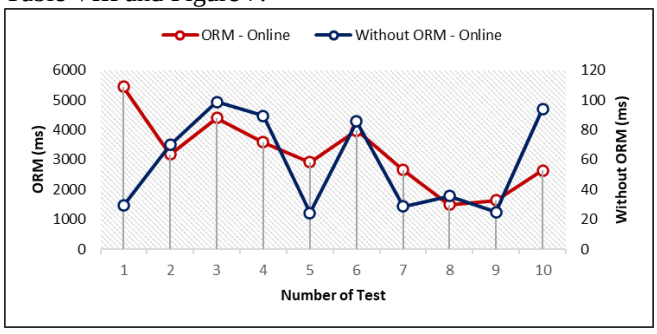

Fig. 7. Performance query One to Many relationship

Table- VIII: Query test for Has Many Through relation

	Online		Offline	
Test	ORM	Non ORM	ORM	Non ORM
1	5440	29.33	84.25	5.56
2	3174.2	70.05	80.87	6.44
3	4392.3	98.72	85.95	5.39
4	3580	89.52	83.16	5.97
5	2920	24.51	84.99	5.54
6	3970	85.56	88.48	4.95
7	2650	28.73	86.75	5.62
8	1490	35.65	83.97	6.52
9	1640	24.83	85.56	5.72
10	2640	93.98	85.5	5.600
Avg.	**3189.65**	**58.088**	**84.948**	**5.731**

From the test in Figure 7, it is known that the Has Many Through relation with the ORM response time online is average 3189.65 ms and with an average time for Non ORM of 58.008 ms to execute the field data.

To get a comparison between offline and online testing Then a summary of the average response time of each relation as seen in Table IX.

Table -IX: Summary offline-online query test relation

	Online		Offline	
Relationship	ORM	Non ORM	ORM	Non ORM
One to One	5142	213.067	140.666	7.105
One to Many	24855	285.294	389.433	21.102
Many To Many	428.552	256.549	12.602	5.666
Has Many Through	3189.65	58.088	84.948	5.731
	8403.801	**203.2495**	**156.9123**	**9.901**

When testing an impedence mismatch case, it occurs when there is a mapping problem in the database relation in displaying details of plant data that displays data from columns with the same name even though the columns are in different tables.

B. Results: Borneo Bioinformatics Optimization

- *PreTest*

The results of the Preliminary Test (PreTest) on the Borneo Bioinformatics portal are presented in Figure 8.

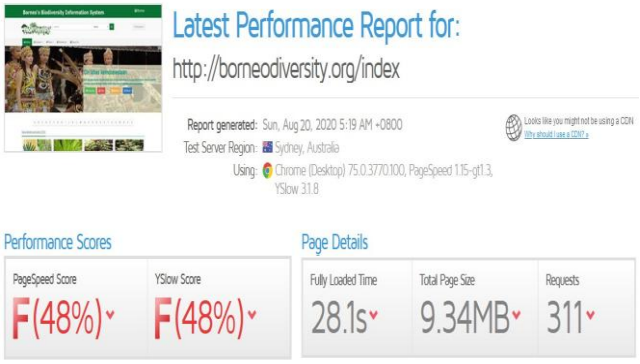

Fig. 8. Screenshot Prestest performance of the portal

The results of the pretest performance reduction obtained a Grade F score (48%) for PageSpeed, and a YSlow's Score with Grade F (48%). For page details (Full Load Time is 28.1 seconds, with a total page size of 9.34MB and 311 Requests).

Figure 9 (pretest) presents a list of recommendations from PageSpeed with the performance scores obtained in the Pre-Test. There are seven (7) recommended items that score very low, i.e. "Serve scaled images, Defer parsing of JavaScript, Optimize images, Inline small CSS, Minify JavaScript" each obtain with Grade F (0). For "Inline small JavaScript" with Grade F(41), Avoid CSS@import with Grade C(73).

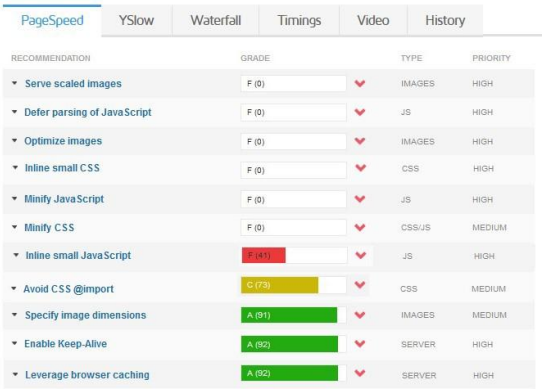

Fig. 9. Screenshoot pretest from google pagespeed

For other recommended items, generally get Grade A with a score of 91 - 100. The low scoring recommendations and optimization solutions from the PageSpeed pre-test results are presented in Table X.

Table X. Performance values of pagespeed recommendations for optimization

Recommendation	Grade Score	Type	Optimization
Serve scaled images	F (0)	Images	Minimized serving scaled images can save 3.3MiB (77% reduction)
Defer parsing of JavaScript	F (0)	JS	Defer parsing JS to reduce blocking of page rendering
Optimize images	F (0)	Images	Optimize images to reduce their size by 965.0KiB (12%)
Inline small CSS	F (0)	CSS	Inlining the response in HTML can reduce-rendering.
Minify JavaScript	F (0)	JS	Minify JavaScript for the resource by 31% reduction)
Minify CSS	F (0)	CSS	Strip unnecessary characters from JavaScript and CSS to speed up download times
Inline small JavaScript	F(41)	Js	Inlining the response in HTML can reduce blocking of page rendering
Avoid CSS@import	C(73)	CSS	The external stylesheets were included in using @import

Whereas for the pretest recommendation from YSlow yahoo is presented in Figure 10.

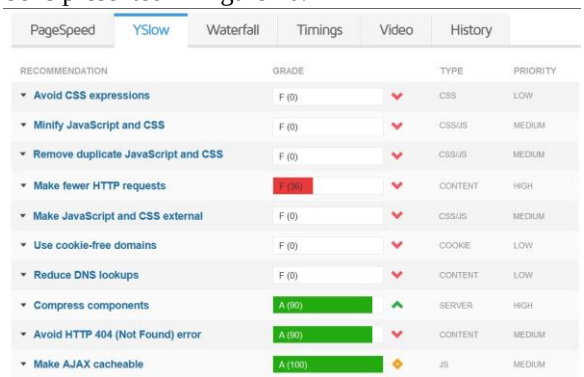

Fig. 10: Screenshot prestest YSlow performance

Figure 10 (prestest) presents a list of recommendations from YSlow with the performance scores obtained in the PreTest. There are four (4) recommended items that score very low, i.e. " Avoid CSS expressions, Minify JavaScript and CSS, Remove duplicate JavaScript and CSS and Make JavaScript and CSS eksternal" each obtain Grade F(0). For other recommended items, generally get Grade A with a score of 90 – 100.

- *PostTest*

The Post-test given to the portal after completing the improvements based on the recommendations of the pretest results (Tabel X), is to measure the performance achievement and optimization of the website.

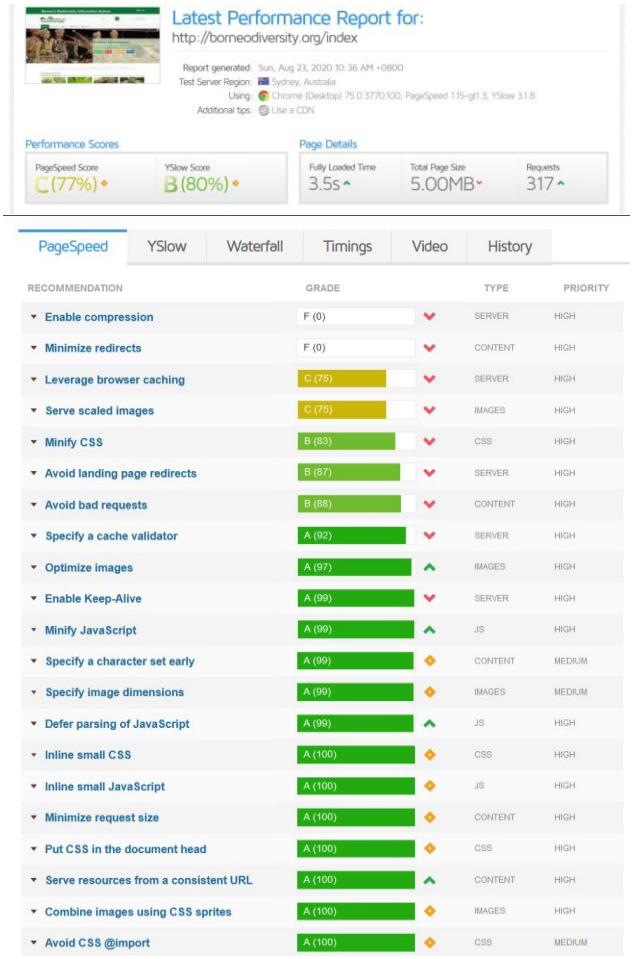

Fig. 11. Screenshot Posttest PageSpeed performance

The performance score of the main portal page after Post-Test gets Grade C (77%) for Pagespeed recommendation, and for YSlow with Grade B(80%). Regarding Detail Pages' performance to get score for Full Load Time is 3.5 seconds, Total Page Size is 9.00MB of 317 Requests. The results of the PostTest measurement in Figure 11 an 12 have shown that the score of each recommendation item has been optimized which shows a good value. For "Serve scaled images with Grade C (75)," Defer parsing of JavaScript with Grade A (99) and "Minify JavaScript with Grade A (99), Optimize images with Grade A(97), Inline small CSS and JavaScript with Grade A(100), Minify CSS with Grade B(83) and JavaScript with Grade A(99). For Avoid CSS@import with Grade A (100).

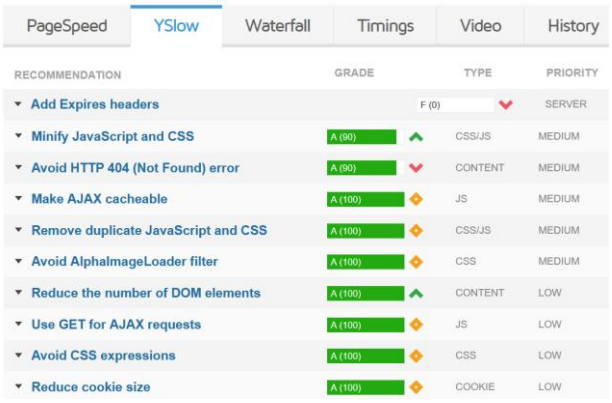

Fig. 12. Screenshot Posttest YSlow performance

Furthermore, the measurement of the Pretest vs PostTest loading times is presented in the Figure 13.

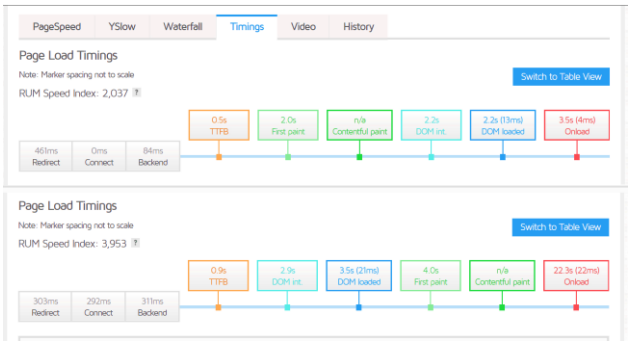

Fig. 13. Screenshot pretest-posttest Page loading timings

In principle, the page loading speed of a website will improve the user experience in accessing information. In principle, the page loading speed of a website will improve the user experience in accessing information. The Bioinformatics portals that reference external objects most of the page load time is spent on HTTP requests separate from images, javascript, and stylesheets. In particular, you can feel the effect of the HTTP request on internet connection and website page load time.

C. Discussion

The bioinformatics portal test is a critical element of SQA and represents a comprehensive review of specifications, design and coding. The test represents an abnormality in the development of the portal. A series of tests systematically reveals several different types of errors.

This study analyzes the performance of the Borneo Bioinformatics portal with a series of tests using measuring and performance optimization tools. Based on the results of testing the response time for online access queries on four database relationship models, an average of 5142 ms is obtained for One to One, One to Many 24,855 ms, Many To Many 428,552 ms and Has Many Through an average of 3189.65. In script testing, image and loading time of portal pages that apply the preTest-PostTest scenario, the optimization of the PageSpeed performance score is F (84%) to C (74%), YSlow from pretest F (84%) to B (80%). Likewise, the page load time, from 28.1 seconds to 3.5 seconds.

Based on the evaluation results, the following were found:

- The query relation model based on the ORM framework makes it easier for database interactions (MVC), but time has a higher response time than without ORM. however, relations without ORM cause impendace mismatch problems in the data mapping process.
- The Efficiency Performance of the Bioinformatics Portal based on the index generated from the PageSpeed Google and YSlow recommendations from Yahoo is still low, and of course it will directly affect the portal's performance.

The results of the evaluation of the bioinformatic portal work in the future require performance improvements and optimization, adjustments to the query relation model, and the appropriate coding structure and serve scale image.

V. CONCLUSION

Indonesia is one of the countries with the highest level of threat to biodiversity [35]. To inhibit the rate of extinction of biodiversity that is accelerated exponentially by human activities, efforts to conserve biodiversity need to be optimized, both in its natural habitat (in-situ conservation) and artificially outside its habitat (external conservation).

Borneo BioInformatics as large-scale system consists of datafields, images and relations, and the data continues to increase and grow. The portal issue is performance optimization. respon time needed to access web pages (page-loading) is still high, becomes slow when accessed, takes up hosting server resources, and other problems.

The results of the study found that the query relation model, parsing script (javaScript and CSS), service scale and dimension images in the interaction process to the database are the dominant resources affecting the performance of the Bioinformatics portal. Performance optimization through determining the appropriate query relation model, minify and defer parsing script or combine images using CSS sprites to reduce scala image.

ACKNOWLEDGMENT

This research was funded by the Higher Education Institution Operational Assistance Fund (BOPTN), Dept. of informatics, Faculty of engineering, Mulawarman University, Samarinda, east-Kalimantan province, Indonesia.

REFERENCES

1. J. Y. Chen, J. V. Carlis, and N. Gao, “A complex biological databases querying method,” *Proceedings of the ACM Symposium on Applied Computing*, vol. 1, no. March 2005, pp. 110–114, 2005, doi: 10.1145/1066677.1066708.
2. A. Messina, A. Fiannaca, L. La Paglia, M. La Rosa, and A. Urso, “BioGraph: A web application and a graph database for querying and analyzing bioinformatics resources,” *BMC Systems Biology*, 2018, doi: 10.1186/s12918-018-0616-4.
3. H. Haviluddin, E. Budiman, and N. F. Hidayat, “A database integrated system based on SOAP web service,” *TEM Journal*, 2019, doi: 10.18421/TEM83-12.
4. A. Kalderimis *et al.*, “InterMine: Extensive web services for modern biology,” *Nucleic Acids Research*, 2014, doi: 10.1093/nar/gku301.
5. E. Budiman, N. Puspitasari, Haerullah, M. Jamil, M. Wati, and A. Saudek, “Evaluation of the bioinformatics resource portal,” in *Proceedings - 2018 3rd International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2018*, 2018, pp. 54–59, doi: 10.1109/ICITISEE.2018.8720973.

6. J. Chomicki and D. Toman, "Temporal Databases," in *Foundations of Artificial Intelligence*, Volume 1., L. V. M. Fisher, D. Gabbay, Ed. Elsevier B.V., 2005, pp. 429–467.
7. P. Cybula, H. Kozankiewicz, K. Stencel, and K. Subieta, "Optimization of distributed queries in grid via caching," 2005, doi: 10.1007/11575863_58.
8. E. Budiman, N. Puspitasari, S. N. Alam, T. M. A. Akbar, Haeruddin, and D. Indra, "Performance analysis of the resource loading time for borneo biodiversity information system," 2018, doi: 10.1109/IAC.2018.8780515.
9. S. Wu, F. Li, S. Mehrotra, and B. C. Ooi, "Query optimization for massively parallel data processing," 2011, doi: 10.1145/2038916.2038928.
10. E. Budiman, N. Puspitasari, M. Wati, J. A. Widians, and Haviluddin, "Web Performance Optimization Techniques for Biodiversity Resource Portal," *Journal of Physics: Conference Series*, vol. 1230, no. 1, 2019, doi: 10.1088/1742-6596/1230/1/012011.
11. L. Zamboulis, N. Martin, and A. Poulovassilis, "Query performance evaluation of an architecture for fine-grained integration of heterogeneous grid data sources," *Future Generation Computer Systems*, 2010, doi: 10.1016/j.future.2010.05.008.
12. E. Budiman and S. N. Alam, "Database: Taxonomy of plants Nomenclature for borneo biodiversity information system," 2018, doi: 10.1109/IAC.2017.8280642.
13. L. Caroprese, E. Zumpano, and E. Vocaturo, "No SQL Database Management Systems for Big Data," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 5, pp. 21–26, 2020, doi: 10.35940/ijeat.D9145.069520.
14. N. Puspitasari and E. Budiman, "Evaluation of Borneo's Biodiversity Information System," *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar, EECCIS 2018*, pp. 434–439, 2019, doi: 10.1109/EECCIS.2018.8692955.
15. N. K. Gundla and Z. Chen, "Creating NoSQL Biological Databases with Ontologies for Query Relaxation," 2016, doi: 10.1016/j.procs.2016.07.120.
16. N. Dengen, E. Budiman, J. A. Widians, M. Wati, U. Hairah, and M. Ugiarto, "Biodiversity information system: Tropical rainforest borneo and traditional knowledge ethnic of dayak," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1–9, 2018.
17. E. Budiman, M. Jamil, U. Hairah, H. Jati, and Rosmasari, "Eloquent object relational mapping models for biodiversity information system," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, Aug. 2017, vol. 2018-Janua, pp. 1–5, doi: 10.1109/CAIPT.2017.8320662.
18. U. Hairah, A. Tejawati, E. Budiman, and F. Agus, "Borneo biodiversity: Exploring endemic tree species and wood characteristics," in *Proceeding - 2017 3rd International Conference on Science in Information Technology: Theory and Application of IT for Education, Industry and Society in Big Data Era, ICSITech 2017*, 2017, vol. 2018-Janua, pp. 435–440, doi: 10.1109/ICSITech.2017.8257152.
19. L. A. Bultet *et al.*, "The SIB Swiss Institute of bioinformatics' resources: Focus on curated databases," *Nucleic Acids Research*, vol. 44, no. D1, pp. D27–D37, 2016, doi: 10.1093/nar/gkv1310.
20. M. Cannataro and P. Veltri, "Bioinformatics web portals," in *Selected Readings on Database Technologies and Applications*, 2008.
21. P. Artimo *et al.*, "ExPASy: SIB bioinformatics resource portal," *Nucleic Acids Research*, 2012, doi: 10.1093/nar/gks400.
22. Haeruddin, H. Johan, U. Hairah, and E. Budiman, "Ethnobotany database: Exploring diversity medicinal plants of Dayak Tribe Borneo," in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017, vol. 2017-Decem, doi: 10.1109/EECSI.2017.8239094.
23. J.-S. Varré, B. Schmidt, S. Janot, and M. Giraud, "Manycore High-Performance Computing in Bioinformatics," 2011.
24. P. D. Karp *et al.*, "A comparison of microbial genome web portals," *Frontiers in Microbiology*. 2019, doi: 10.3389/fmicb.2019.00208.
25. W. W. Li *et al.*, "Building cyberinfrastructure for bioinformatics using service oriented architecture," 2006, doi: 10.1109/ccgrid.2006.1630932.
26. E. Budiman and S. N. Alam, "User perceptions of mobile internet services performance in borneo," in *2017 Second International Conference on Informatics and Computing (ICIC)*, Nov. 2017, vol. 2018-Janua, pp. 1–6, doi: 10.1109/IAC.2017.8280643.
27. S. Kaur, K. Kaur, and P. Kaur, "An Empirical Performance Evaluation of Universities Website," *International Journal of Computer Applications*, 2016, doi: 10.5120/ijca2016910922.
28. D. E. Luna, L. F. Luna-Reyes, J. R. Gil-Garcia, and R. Sandoval-Almazán, "Government web portals performance evaluation using data envelopment analysis," 2011, doi: 10.1145/2037556.2037617.
29. D. google, "About PageSpeed Insights," *developers.google.com*. https://developers.google.com/speed/docs/insights/v5/.
30. Marcelduran, "Web Performance Best Practices and Rules," *yslow.org*. http://yslow.org/.
31. Carbon60, "Recommendations," *gtmetrix.com*. https://gtmetrix.com/recommendations.html (accessed Jun. 06, 2020).
32. marcelduran, "YSlow Ruleset Matrix," *yslow.org*. http://yslow.org/ruleset-matrix/.
33. Widjaja, *Kekinian Keanekaragaman Hayati Indonesia 2014*. 2014.
34. E. Budiman, N. Puspitasari, M. Wati, Haviluddin, and R. Rahim, "Model Framework for Development of Biodiversity Information Systems," *Journal of Physics: Conference Series*, vol. 1230, no. 1, 2019, doi: 10.1088/1742-6596/1230/1/012012.
35. D. F. R. A. Cleary and L. DeVantier, "Indonesia: Threats to the Country's Biodiversity," *Encyclopedia of Environmental Health*, no. November 2017, pp. 622–632, 2011, doi: 10.1016/B978-0-444-52272-6.00504-3.

AUTHORS PROFILE

Edy Budiman is member of the Association for Computing Machinery (ACM), member of Institute of Electrical and Electronics Engineers (IEEE), and member of APTIKOM (Asosiasi Pendidikan Tinggi Informatika dan Komputer) and mem-ber of The Institution of Engineers Indonesia (PII). Currently, he is actively teaching and researching. As a writer on several journals and conferences, he focuses his research on mobile network issues, performance and mobile-based apps.