Page Load Time Speed Increase on Disease Outbreak Investigation Information System Website

Rahmat Oktrifianto[1], Dani Adhipta[2], Warsun Najib[3]

***Abstract*—Outbreaks or extraordinary events often become an issue that occurs in Indonesia. Therefore, an outbreak investigation information system is required to collect, manage and analyze data quickly and accurately. On the other hand, challenges in data accessing processes in certain locations are still constrained by a slow internet connection. This paper conducted speed increase of a page load or site speed time from disease outbreaks investigation information system website. Page load time speed testing was carried out using Google Chrome Developer Tools and using simulation speeds of 2.5 Mbps. Testing time was carried out by dividing the time into three sections, morning hours, working hours and night hours. Implementation of page load time increase includes reducing HTTP requests, utilizing GZIP compression, performing code minification, setting browser chache, using CDN, and using other enhancement techniques. The results showed that after implementing an increase in page load time by turning off cache and using cache, there was an increase in site speed. When the browser cache was turned off, an average page load time increased of 54.79% from the previous time. Whereas when using the browser cache, page load time speed increased by 55.28% from the previous time.**

***Keywords*—information system, outbreak, page load time, website, site speed.**

I. Introduction

Nowadays, development of information technology world takes place very quickly. The internet has become an important part of life in helping various human works. Every day, internet users in Indonesia continue to increase. Based on research from We Are Social and Hootsuite in 2017, internet users in Indonesia increased by 51% in one year [1]. A high growth of the internet certainly has an impact on progress of the Indonesian nation, including in the health world.

On the other hand, outbreaks transmission or extraordinary events often become hot news in Indonesia. Various infectious diseases can rapidly develop without being quickly identified [2]. In 2014, Ministry of Health of Indonesia reported that the use of smallpox vaccine took in 94.7% of children, but only 69.5% in South Kalimantan. Although there are efforts to vaccinate in Indonesia, from 2013 there were 11,521 smallpox cases and in 2014 there was an increase to 12,943 smallpox cases [3]. To overcome problems in monitoring and analyzing the spread of extraordinary events, an information system capable of processing information is required, both through websites and mobile applications.

Indonesia remains a developing country and still continues to improve in development of all fields, especially in internet technology field. Indonesia's internet access speed is still relatively very slow compared to other countries. Based on Akamai data, in the second quarter of 2014 Indonesia's internet speed was ranked 101st. The average internet access speed was around 2.5 Mbps, so it was still far behind other countries [4]. Of course this slow internet access causes many obstacles in accessing outbreak investigation information systems through the website. Rapid site process is required to analyze and collect data quickly and accurately.

The page load time speed test results of disease outbreak investigation information system at Faculty of Medicine, Public Health, and Nursing UGM showed an average speed of 5.121 seconds and were classified as slower than other websites which had an average speed of 2.648 seconds. Therefore, it is necessary to increase the page load time speed. For this reason, development of a fast information system website needs to be implemented to collect data, analyze and process data to provide useful information for Indonesian people. With the presence of this paper, it is expected that development of disease outbreak investigation information system website on page load time performance will improve. Page load time speed increase is expected to improve levels of trust and satisfaction as well as increase user experience for information system users.

II. Page Load Time

Page load time is a required time to download and display all content from a web page in a browser [5]. In [6], an increase in speed on front-end part such as applying GZIP compression, performing minification code, setting browser cache optimization, using content delivery network, and others were carried out. This study results showed an initial increase in page load time speed of 29.7%, from 1.82 seconds to 1.28 seconds. This study results did not only affect page load time speed of one page but it also affected total bandwidth and transfer process from web server.

There was another research on page load time speed increase [7]. This research aimed to compare loading speed from two websites containing both image and text data. This research focus was to increase page load time using image compression techniques, minification code, external CSS and JavaScript files combination, and asynchronous JavaScript. From this study results there was a speed increase of 24.57% when implementing an increase in page load time, from 362.2 milliseconds to 273.2 milliseconds.

[1] *Department of Electrical and Information Engineering, Faculty of Engineering, Universitas Gadjah Mada, Jln. Grafika 2, UGM Yogyakarta 55281 INDONESIA (phone: 0274-555 305; fax: 0274-552305; e-mail: rahmat.oktrifianto@mail.ugm.ac.id)*

[2,3] *Lecturer, Department of Electrical and Information Engineering, Faculty of Engineering, Universitas Gadjah Mada, Jln. Grafika 2, UGM Yogyakarta 55281 INDONESIA (telp: 0274-552305; fax:0274-552305; e-mail[2]: dani@te.ugm.ac.id; e-mail[3]: warsun@ugm.ac.id)*

Outbreak or Extraordinary Events (*Kejadian Luar Biasa*/KLB) is an event that exceeding normal expectations of an illness related to an event and clearly determined in a community, region, and at a certain time [8], [9]. According to the Indonesian Ministry of Health, extraordinary events are occurrence or increase in incidence of morbidity and/or death that is epidemiologically meaningful in an area within a certain period of time and is a condition that leads an outbreak occurrence [10].

A study evaluated the surveillance system based on Early Warning and Reporting System (EWARS) in an effort to detect outbreaks in Barru District [11]. Processing evaluation and analysis of extraordinary events data was carried out using coding, while data collection was carried out by interviews and in-depth observations. This research was still focused on evaluating inputs, processes, outputs, and outcomes from EWARS-based information systems implementation. Evaluation of this study had not provided and developed an information system that was managed rapidly using a website or application basis. However, in its implementation, information system evaluation had been able to detect infectious diseases quickly so that it could reduce mortality compared to a period before EWARS implementation.

In [12] a development on geographic information systems application was carried out to map out disease outbreaks and immunization coverage in the city of Yogyakarta. The main focus of this research was to find out distribution patterns of outbreak-based diseases and immunization coverage of health center in Yogyakarta. This study still did not provide specifications regarding applications and websites because it focused on developing geographic information systems. In addition, this information system development was still limited to certain regions.

Page speed, or commonly called site speed, is a speed of displaying a page from a website. Page speed is usually described as page load time or time to first byte (TTFB) or page response time, but there are no specific terms to describe it. Page load time is time required to display full page contents, while time to first byte is time required by a browser to receive first byte information from web server [13]. Page response time is time requests to send all data from one server to client browser [14].

Node.js is a software written in JavaScript programming language and it is commonly used for web application development. Node.js was first created by Ryan Dahl in 2009 and he made JavaScript worked on server side. Node.js requires an engine to run on server side, namely the V8 engine which is engine on Google Chome browser created by Google [15]. Express.js is a framework from Node.js that uses JavaScript programming language. Express.js was first developed by TJ Holowaychuk, but is now managed by a community.

Chrome DevTools is a tool used for debugging in Google Chrome browser. DevTools provides access for website developers to view a web application source code in a browser. DevTools is used in debugging layouts, managing JavaScript, and for viewing code from a web page [16].

III. Method

A. Materials and Research Tools

The material used in this study was a source code of disease outbreak investigation information website system in a form of HTML files, EJS, stylesheets, and JavaScript, while research tools consisted of hardware and software. For hardware used in this research were laptops with Windows 10 operating system with Intel® Core ™ processor specifications i3-4030U @ 1.90 GHz (4 CPUs), 10 GB DDR3 RAM memory, 120 GB SSD capacity, 500 GB HDD capacity, and VPS Server. Node.js, WebStorm, Visual Studio Code, and Google Chrome Developer Tools were softwares used in this research.

B. Research Stream

This study conducted a development of page load time speed increase, especially in front-end part of website information system. Measures taken included literary study, system need analysis, website development, implementation of page load time speed increase, testing, and final report writing. This research stream can be described as in Fig. 1.

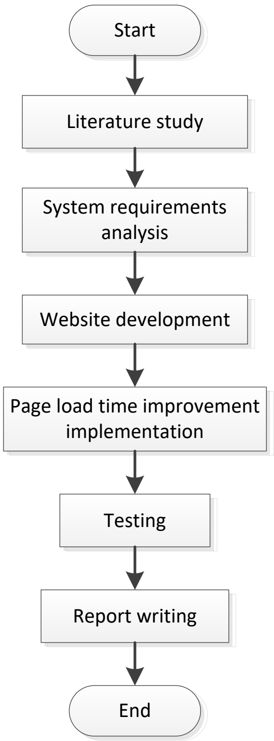
Start
Literature study
System requirements analysis
Website development
Page load time improvement implementation
Testing
Report writing
End

Fig. 1 Research stream.

C. Systems Needs Analysis

Before performing program code writing, it was necessary to know design concept of system requirements to facilitate interface development in this study. System interface development itself was connected through website and Android smartphones. Data functionality section was connected to database through Application Program Interface (API), as shown in Fig. 2.

Based on needs, that information system was used for administrators, investigators, and user as supporters in data collecting as in Fig. 2. Administrators were staffs appointed to monitor all system conditions. Investigator was a staff appointed to review outbreak cases occurrences and to provide input regarding further actions in the event of a case. Users were registered users and could create a project to report if an outbreak case occured. Each actor had a different role from one another.

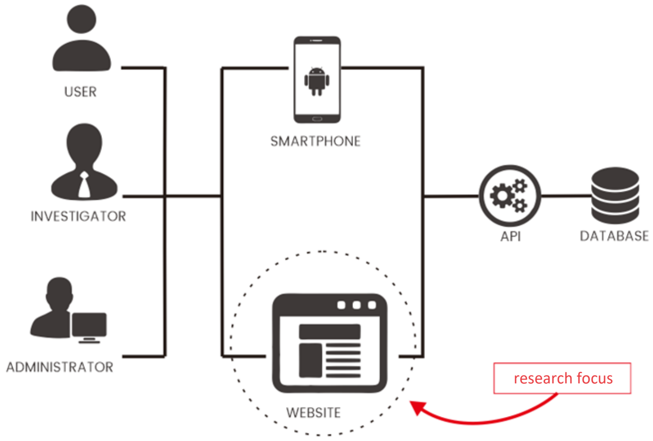

Fig. 2 System Needs Analysis.

This outbreak investigation information system was divided into two, i.e., website and Android application, as illustrated in Fig. 2. However, this development of page load time increase only focused on website section. In addition, this speed increase also did not include API section and database because they have been developed in project development team.

D. Website Development

Website interface of this disease outbreak investigation information system was developed using HTML, CSS, and JavaScript. This website application was constructed using Wxpress.js framework by employing Node.js back-end system. However, back-end section of REST API and this information system database had been developed.

In order for page interface to be more responsive and neat, the Materialize CSS framework was also used. Furthermore, to assist in program code simplification, templating Embedded JavaScript (EJS) was employed. To facilitate development locally, WebStorm IDE was employed to collaborate with other application developers.

E. Increase Implementation

At this stage, steps for implementing a disease outbreak information system aiming to increase website's page load time speed and to reduce resources needed to load an information system website page are explained.

1) Reducing HTTP Request: Reducing requests is an option in increasing page load time speed. It is because a smaller browser request to web server will speed up data transfer. HTTP requests can be reduced by combining JavaScript files, combining CSS files, reducing unnecessary images, and others. In this implementation, codes of before and after implementation still used same or slightly different requests number for page equalization.

2) GZIP Compression: Each website page has a collection of static files such as HTML, CSS, JavaScript and several other files. Compression makes file size becomes smaller so that it increases page loading speed. This GZIP compression was done in back-end section of the Express.js application by adding compression modules to Express.js application through package manager node.

3) Code minification: Deleting some unused program codes, suchas using white space, comments, and blank lines will make file size get smaller. Code minification can be done with Notepad++ software or using many tools available on the internet. This code minification was carried out in HTML, CSS, and JavaScript code files. Online tools employed in this study were http://minifycode.com, which was free and could be used easily because it only had to copy and paste the code.

4) Setting Cache: Setting the expiration time of cache to store static files on browser affects many file downloads from server. More files stored on website visitor local disk will reduce server load and speed up page loading. In this study, cache expiration time was activated for 1 month or 2,628,000 seconds. To set cache time to expire, "max-age code command" [time in seconds]" in Cache-Control section of Express.js app.js file was used.

5) Using CDN: A closer access distance with server location will reduce response time so that it accelerates page loading time. This study employed CSN process by Cloudflare to retrieve static files such as CSS and JavaScript libraries. CDN used in handling these CSS and JavaScript files was free of charge.

6) Image Optimization: Image optimization is very important because an appropriate image size and format speeds up downloading image content process thereby reducing page load time. However, this study still focused on compression section to reduce file size. To do image compression, an available online tool, namely "TinyJPG", which could be accessed from https://tinyjpg.com was utilized . To use this tool just drag and drop images then do image compression.

7) Other Improvement Techniques: There are also various other simple ways to increase page load time speed of a website page. First, repairing broken links or 404 (not found) could be a factor that increased page loading speed. To view damaged pages, various online tools such as Google Webmaster Tools could be used. Second, by reducing redirect link to a particular page so that it would automatically reduce HTTP Request. Third, by declaring character type on HTML page using "UTF-8" type in <head> section. Fourth, by declaring meta viewport in <head> section, because it would affect rendering speed in mobile view. Fifth, by putting CSS structure for top and JavaScript at bottom part inside <body>. This was due to performing elements *rendering* process at first and then running a *script* to speed up *page load* time of a *website* page.

F. Testing

This test was carried out using Google Chrome Developer Tools browser. Page load time speed was measured in units of seconds. Page load time speed testing on a web page could utilize network panel feature on Chrome DevTools. Tests were carried out on every website page, then page load time speeds between version 1 program and version 2 program codes were compared. The first version was a website before implementation of an increase in page load time, while the second version was a website after an implementation of an increase in page load time.

The testing time was carried out into three different times, namely morning hours, working hours, and night hours, as shown in Fig. 3. This was because outbreak investigation information system was a website application that was always online and realtime for 24 hours. Even though it was carried out at different times, but in testing it was still employed a same internet access speed simulation, which was 2.5 Mbps with 300 ms latency. This was because the average internet speed in Indonesia was 2.5 Mbps so that the speed was selected [4].

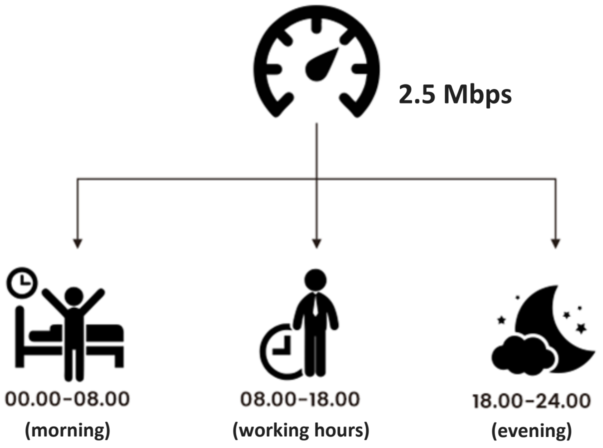

Fig. 3 Time scheme of page load time testing.

First, testing was carried out at dawn until morning when internet traffic was very tenuous, between 00.00 WIB - 08.00 WIB. Second, testing was carried out during working hours or during heavy internet traffic, which was around 08.00 WIB - 18.00 WIB. Third, testing was carried out at night when internet traffic was not too crowded, which was between 18.00 WIB - 24.00 WIB. Of those three testing times, average page load time speed of each page was obtained.

IV. Results and Discussion

After developing and implementing page load time on outbreak investigation information system website, testing was carried out. Page load time speed testing could be divided into before and after increase implementation. Of those two distinguishing factors, each was distinguished in both disable cache and enable cache.

The use of disable cache was a speed test when website was first loaded and had never downloaded any cache from server, while enable cache was assumed to be testing speed when cache was stored in browser's local storage or when in reloaded states. To activate and deactivate browser cache, settings on network tab in Google Chrome DevTools were used.

A. Comparison by Turning Off Browser Cache.

Turning off the browser cache or disabling cache in browser was intended to prevent local cache retrieval that had been stored. From Table I, all pages have decreased time or increased page load time by an average of 54.79% from original time.

Table I
Page Load Time Testing Results Before and After Increase Implementation by Turning Off Browser Cache

Page	Before	After	Time Reduction
Login	3.33 s	1.073 s	67.77%
Register	3.14 s	0.821 s	73.84%
Forgot Password	6.097 s	2.59 s	57.52%
Dashboard	6.683 s	3.847 s	42.44%
Single Project	7.083 s	4.207 s	40.61%
Create Project	4.920 s	2.257 s	54.13%
Create Report	4.893 s	2.3 s	53.00%
Custom Form	4.88 s	2.25 s	53.89%
List of Investigators	5.063 s	2.373 s	53.13%
List of Users	5.117 s	2.477 s	51.60%
(disable cache) Average Decrease (%)			54.79%

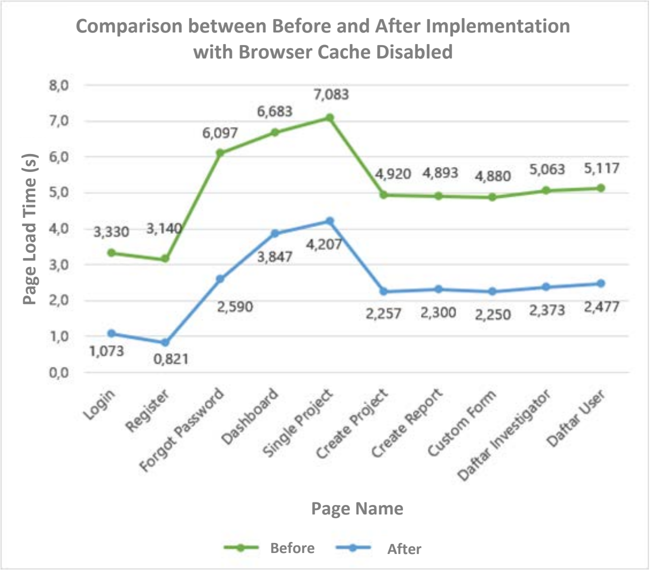

Fig. 4 Comparison graph of before and after increase implementation by turning off browser cache.

It can be seen that after implementation there is a decrease in time or an increase in page load time speed of all pages, as shown in Fig. 4. The largest number of increases occurred on Register page, which amounted to 73.84% of original time, while the smallest number of increases occurred on Single Project page, which amounted to 40.61% of original time.

B. Comparison by Turning on Browser Cache.

Turning on browser cache or enable cache in browser had a purpose to retrieve cache that had been stored in local memory, thereby reducing download load from internet. From Table II it can be seen that on all pages there is also a decrease in time or increase in page load time when turning on browser cache with an average increase of 55.28% from original time.

TABLE II
PAGE LOAD TIME TESTING RESULTS BEFORE AND AFTER INCREASE IMPLEMENTATION BY TURNING ON BROWSER CACHE

Page	Before	After	Time Reduction
Login	0.746 s	0.367 s	50.74%
Register	0.574 s	0.358 s	37.67%
Forgot Password	3.713 s	0.871 s	76.55%
Dashboard	1.837 s	0.411 s	77.64%
Single Project	1.683 s	1.143 s	32.08%
Create Project	0.894 s	0.427 s	52.22%
Create Report	0.922 s	0.416 s	54.86%
Custom Form	0.900 s	0.390 s	56.67%
List of Investigators	0.895 s	0.390 s	56.42%
List of Users	0.919 s	0.387 s	57.93%
(enable cache) Average Decrease (%)			55.28%

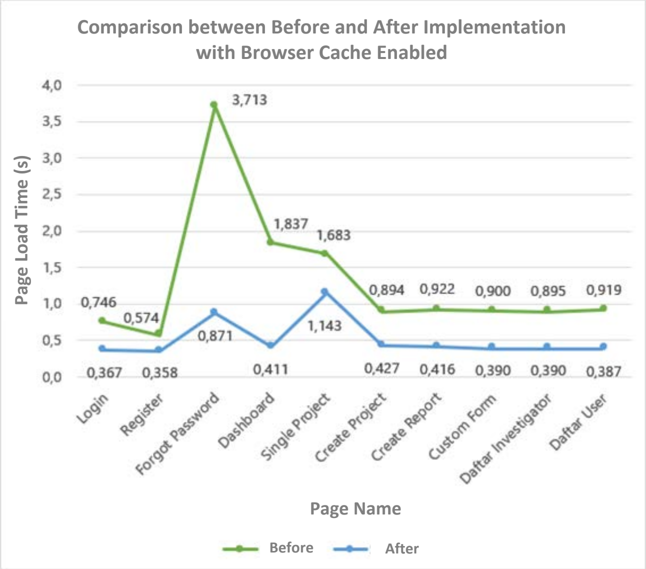

Fig. 5 Comparison graph of before and after increase implementation by turning on browser cache.

It can be seen as in Fig. 5 that cache use is very influential in transferring data of a page file size. When browser cache was activated, the highest increase in page load time speed occured on Forgot Password page, from 3.713 seconds it changed to 0.871 seconds, while the smallest increase in page load time speed occured on Register page, which was from 0.574 seconds to 0.358 seconds.

V. CONCLUSIONS

A. Conclusions

From testing that has been done, conclusions can be taken as follows. The use of minification code could reduce stylesheet file size by 22.35% from its original size, which was 179 kB to 139 kB. Whereas for JS files, its size could be reduced by 68.05%, from original size of 266 kB to 85 kB. This is because minification process can reduce various characters types that are not used in code writing. The use of GZIP compression in this study could reduce file size by 76.84% from its original size, i.e. from 368 kB to 85.2 kB after compression. Then, page load time test results before and after increase implementing by turning off browser cache (disable cache) shows an increase in average page load time speed of 54.79% from previous time. In addition, average file size decreases by 63.01% from its original size. Turning off browser cache aims to prevent cache data from being stored locally. The results of testing page load time before and after implementation using browser cache (enable cache) indicate an increase in average speed page load time of 55.28% from previous time. In addition, average file size decreases by 71.29% from its original size. Turning on browser cache aims to retrieve cache data that has been stored in local storage thereby reducing data transfer from the internet.

B. Suggestions

For further development, testing can be carried out in different places, considering that this information system is used in locations with different internet conditions. In addition, it is also necessary to develop an increase in page load time automatically.

REFERENCES

[1] (2017) "Pertumbuhan Pengguna Internet, Indonesia Nomor 1 di Dunia," [Online], http://databoks.katadata.co.id/datapublish/2017/05/22/pertumbuhan-pengguna-internet-indonesia-nomor-1-di-dunia, access date: 30 Oct. 2017.

[2] M.C. Schraefel and E.F. Churchill, "Wellth Creation: Using Computer Science to Support Proactive Health," *IEEE Journals & Magazines*, Vol. 47, No. 11, pp. 70-72, 2014.

[3] E. Hartoyo, A. Wiyatno, U.A. Jaya, C.N. Ma'roef, C. Monagin, K.S. Myint, and D. Safari, "Occurrence of Measles Genotype D8 During a 2014 Outbreak in Banjarmasin, South Kalimantan, Indonesia," *International Journal of Infectious Diseases*, Vol. 54, pp. 1-3, 2017.

[4] D.W. Ningrum (2014) "Kecepatan Internet Indonesia Masuk Kategori Paling Lambat," [Online], http://tekno.liputan6.com/read/2113835/kecepatan-internet-indonesia-masuk-kategori-paling-lambat, access date: 30 Oct. 2017.

[5] A. Ndegwa (2016) "What is Page Load Time?," [Online], https://www.maxcdn.com/one/visual-glossary/page-load-time/, access date: 4 Nov. 2017.

[6] I. Illiev and G.P. Dimitrov, "Front End Optimization Methods and Their Effect," *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2014, pp. 467-473.

[7] P.H. Shroff and S.R. Chaudhary, "Critical Rendering Path Optimizations to Reduce the Web Page Loading Time," *2nd International Conference for Convergence in Technology (I2CT)*, 2017, pp. 937-940.

[8] R.K. Gupta, P. Dudeja, and A.S. Minhas, "Chapter 3 – Outbreak Investigation of Foodborne Illnesses," in *Food Safety in the 21st Century*, Elsevier, 2017, p. 29–40.

[9] A.R. Datta and L.S. Burall, "Serotype to Genotype: The Changing Landscape of Listeriosis Outbreak," *Food Microbiology*,Vol. 75, pp. 18-27, 2017.

[10] J. Iswanto (2012) "Wabah/KLB Penyakit Menular," [Online], http://www.sumbarsehat.com/2012/08/wabahklb-penyakit-menular.html. access date: 18 Nov. 2017.

[11] M. Arsyam and H. Kusnanto, "Evaluasi Penerapan Sistem Informasi Surveilans Berbasis Ewars dalam Upaya Deteksi Dini Kejadian Luar Biasa di Kabupaten Barru," Master thesis, Universitas Gadjah Mada, Yogyakarta, Indonesia, 2013.

[12] I. Rasyitaningrom and P. Widayani, "Aplikasi Sistem Informasi Geografi untuk Pemetaan Kejadian Luar Biasa Penyakit dan Cakupan Imunisasi di Kota Yogyakarta Tahun 2014," Bachelor thesis, Universitas Gadjah Mada, 2014.

[13] (2017) “What is Page Speed?,” [Online], https://moz.com/learn/seo/page-speed, access date: 7 Jan. 2018.

[14] S. Long (2014) “What exactly is Page Response Time?,” [Online], https://stackoverflow.com/questions/21268328/what-exactly-is-page-response-time, access date: 13 Jan. 2018.

[15] F. Lutfi (2017) “Mengenal Node.js,” [Online], https://www.codepolitan.com/mengenal-nodejs-5880234fe9ae3, access date: 13 Jan. 2018.

[16] (2017) “Google Chrome Developer Tools,” [Online], https://developer.chrome.com/devtools#further-reading, access date: 4 Nov. 2017.