

Application of Progressive Web Apps (PWA) on PT SKA's E-Commerce Website

Evangeline Eunike¹, Ridwan Sanjaya², Albertus Dwiyoga Widiatoro³

^{1,2,3} Department of Information Systems, Computer Science Faculty Soegijapranata Catholic University

^{1,2,3}Jl. Pawiyatan Luhur IV No.1, Bendan Duwur, Kota Semarang, Jawa Tengah 50234

¹19n40001@student.unika.ac.id

²ridwan@unika.ac.id

³yoga@unika.ac.id

Abstract— Mobile app developers are often overwhelmed when they must create mobile applications, which must create applications for iOS and Android users. The problem is that each one has a different programming language, for iOS users it must be developed with Objective-C or Swift while for Android it is developed with Kotlin or Java programming language. This research was conducted with a tool that in the research process using observation. Observations were made by testing native applications with applications that were already using Progressive Web Apps (PWA) using the PageSpeed Insights tool. Progressive Web Apps (PWA) help optimize websites so that they look like native applications both on computers and on mobile. Application can be accessed offline, provide notifications, and there is no need to download application on the PlayStore or App Store, just through the website. This PWA exists to address these issues. PWA makes things faster and optimizes quality.

Keywords— e-commerce, mobile, progressive, push notification, PWA, responsive, website

I. INTRODUCTION

In the process of making application often overwhelmed mobile app developers, where must make application for iOS and Android users [1]. The problem is that each one has a language different programming, for iOS users must developed with Objective-C or Swift whereas for Android developed with

Kotlin or language Java Programming [2]. Type Applications also vary, including:

The first is native mobile apps. Just as the name means it's the original form of an application, be it an Android application it's both a native application which means for Android it is necessary to make one application itself and for iOS one need to make another application itself [3]. If want make application for both platforms like for Android and iOS, then must make two application native, one application for Android and one for iOS. For example, there is a little desired changes to the application, then must to do changes on Android as well as iOS. Android apps usually use language Java programming, while iOS uses Objective-C [4].

Second, web application. This means an application that runs on the web, where the application can run with the help of a web. It is like saying there is an E-Commerce web and then want to make a replica but the mobile version can use what is called a web application. Whatever changes are made on the web will also change on the mobile display [5].

Third, hybrid apps. This application will run on the Android platform as well as on iOS. Only need to create one file and it will run on both Android and iOS. There is no need to create a programming language for Android itself, iOS itself. Enough with one program that can run on both platforms [6].

Fourth, Progressive Web Apps. With Progressive Web Apps (PWA), which was previously a website, it can appear like a native mobile application. So, the website can

be accessed offline, provide notifications, and can install the website on mobile. Users also do not need to download applications on the Play Store or App Store, because basically the application created is a website that uses Progressive Web Apps (PWA) technology so that the website runs like a native mobile application in general [7]. When accessing the Progressive Web Apps (PWA) application, it does not matter whether it is Chrome or Opera browser. Whether it is accessed from Indonesia or America, it can still be accessed with fast loading times. Progressive Web Apps (PWA) will work well because the core is built with progressive improvements [8].

Progressive Web Apps (PWA) help optimize websites so that they look like native applications both on computers and on mobile. The difference between native applications and applications that use Progressive Web Apps (PWA) is when running the application. If want to use the original application, an internet connection (online) is required and a large enough memory. While applications that use Progressive Web Apps (PWA) can be used without an internet connection (offline) by consuming less memory [9]. Following are the features of PWA:

Service Worker, with this service worker feature, users can open applications offline. Technically, this service worker acts as a “network proxy” that mediates between the browser and the server [10].

Creating Shortcuts on Homescreen, if the native mobile application is usually installed via the PlayStore or App Store. Once installed, the application will appear on the homescreen along with the logo of the application [11].

Application Shell, when open a native mobile application, there is usually a splash screen for 1 or 2 seconds, then can see the main page of the application. With the Application Shell data will load quickly because the focus is on speed and what is updated on the website will be update in the application [12].

Transport Layer Security, compared to native mobile applications, Progressive Web

Apps (PWA) can help an application increase security so that the application becomes more secure. In making applications using Progressive Web Apps (PWA), HTTPS is used as a protocol to make secure connections [13].

Web App Manifest File is a part if Progressive Web Apps. The JSON file owned by the Web App Manifest contains information consisting of the website name, icon, size used, default color, start page and splash screen. Such information can be entered directly [14].

Push Notifications, here Push Notifications will run like native applications in general. A notification will appear on the user's screen even though he is not opening the application [15].

Linkable, just like websites in general that use URLs to become characteristics. When want to share a Progressive Web Apps (PWA) application that is being used with others, simply copy the existing URL. URLs can be easily shared with others without a complicated installation process [16].

For this reason, in this thesis, the solution for developing PWA based applications needs to be investigated further. So that it is known the strategy of implementing Progressive Web Apps (PWA) on E-Commerce Websites.

II. METHOD

This research method is carried out with measurable tools, which in the research process uses observation. Observations were made to make a comparison between the two applications that have not used Progressive Web Apps (PWA) and after using Progressive Web Apps (PWA) using PageSpeed Insights. PageSpeed Insights is a tool from Google that can be used to check the speed of a website page on mobile and desktop displays.

Based on the parameters referenced by Google Developers to rate a website, PageSpeed Insights categorizes the scoring rating from 0 to 100 to determine whether the website is performing well or not based on the existing parameters.

Table 1. Score Category in Web Scoring

Score	Status	Color Status
0-49	Not good (usually slow)	Red
50-89	Standard	Orange
90-100	Very Good (usually fast)	Green

The categories of scores can be seen in Table 1. Table 1 shows that if a web after being tested gets a score of 0-49, then the status of the web is slow or not good. If the web after being tested gets a score of 50-89, then the status of the web is standard or average like the web in general. If a web after being tested gets a score of 90-100, then the status of the web is fast or very good. The colors from slow to fast states are red, orange, and green.

Table 2. Percentage Weight

Number	Condition	Weight
1	First Contentful Paint	10%
2	Speed Index	10%
3	Largest Contentful Paint	25%
4	Time to Interactive	10%
5	Total Blocking Time	30%
6	Cumulative Layout Shift	15%

Table 2 shows the performance scores obtained on mobile and desktop displays are influenced by six analyses, namely:

First Contentful Paint, how much time it takes to display an image or text the first time. First Contentful Paint weight is 10%.

Speed Index, how fast a page is clearly visible. The weight for the Speed Index is 10%.

Largest Contentful Paint, indicates the load time required to render the largest elements of a web page including images and text. Weight of Largest Contentful Paint by 25%.

Time to Interactive, the time it takes for the page to be considered interactive overall. The weight for Time to Interactive is 10%.

Total Blocking Time, the total amount of time between First Contentful Paint and Time to Interactive. Weight for Total Blocking Time is 30%.

Cumulative Layout Shift, assessing the visual stability of a web page, this analysis will calculate the total number of times the web page shifts. The lower the value, the sign of a slight shift that has occurred. Weight Cumulative Layout Shift by 15%.

III. RESULTS AND DISCUSSION

A. RESULT

The test was carried out on one of the E-Commerce websites, namely <http://testing.ska-indonesia.com/wp-admin/> of 132 megabytes.

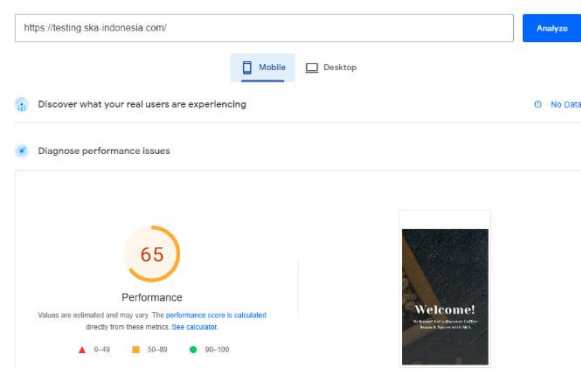


Figure 1. Performance Without PWA on Mobile

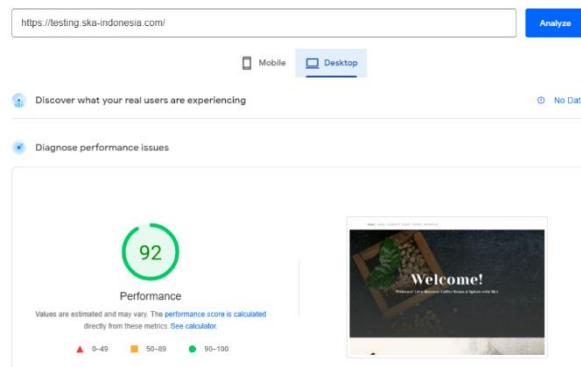


Figure 2. Performance Without PWA on Desktop

From the test results without using Progressive Web Apps (PWA), it can be seen in Figure 1 that the overall performance result is 65% on the mobile display. While the results in Figure 2 show a figure of 92% for the desktop display.

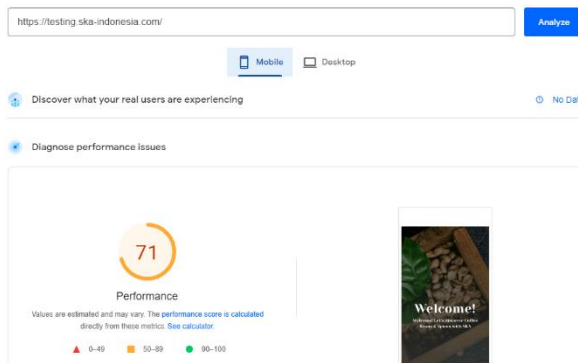


Figure 3. Performance With PWA on Mobile

The performance results in Figure 3 obtained for mobile displays that are already using Progressive Web Apps (PWA) are 71%. Online a 6% difference from the results before using Progressive Web Apps (PWA) on a mobile display.

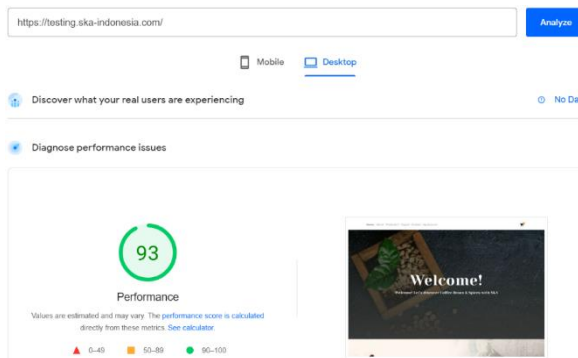


Figure 4. Performance With PWA on Desktop

The result shown in Figure 4, namely the performance for desktop displays that already use Progressive Web Apps (PWA) get 93% results. When compared with the previous results, it is surprising, because the difference is only 1%.

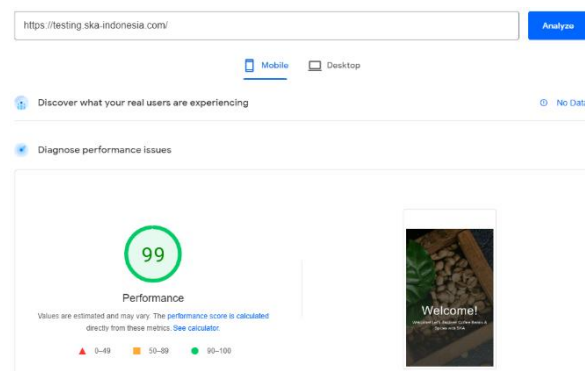


Figure 5. Performance After Compressing Image and Reduce of Code on Mobile

In Figure 5 above, there is a significant difference in the almost perfect mobile display with a total value of 99%.

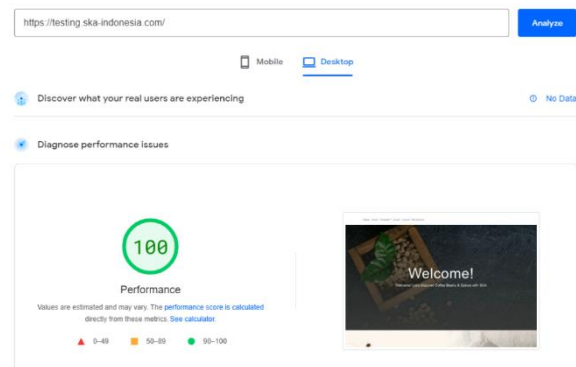


Figure 6. Performance After Compressing Image and Reduce of Code on Desktop

In Figure 6, the performance after compressing the image and removing lines of code that does not need to be obtained is 100%.

B. DISCUSSION

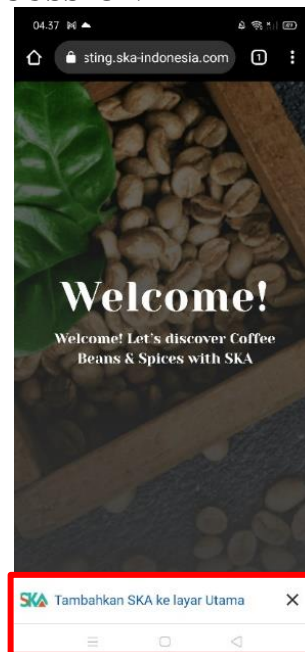


Figure 7. Features Add to Homescreen

After the web has successfully implemented PWA, when accessing the website, a banner will appear to add applications to the mobile device as shown in Figure 7.

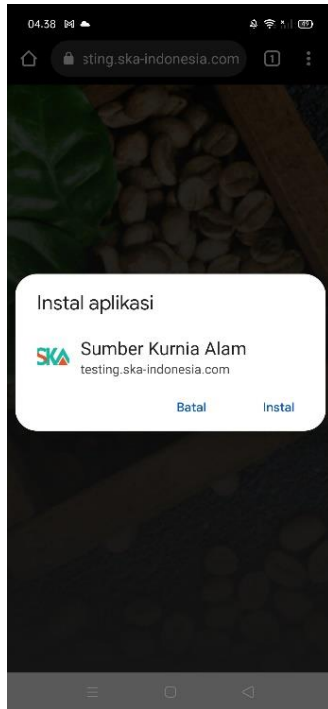


Figure 8. Install Process

Then the banner is clicked, a confirmation question will appear whether you are sure you want to add the application to the mobile, this can be seen in Figure 8.



Figure 9. Homescreen Shortcuts

If have added a web application to your device, an icon for the application will appear as shown in Figure 9. The presence of an application icon that appears further proves

that the homescreen shortcut creation featured on the PWA is valid according to its function.



Figure 10. Splash Screen

When open the application, the splash screen will appear like the native and web application has successfully implemented the Application Shell feature, this can be seen in Figure 10.



Figure 11. Service Worker

The web application can also run offline, as evidenced by the PWA component featured,

namely the Service Worker, this can be seen in Figure 11.



Figure 12. Push Notification

Finally, there is a Push Notification in Figure 12 which show that the PWA web application can send a notification.

IV. CONCLUSION

There is a change in performance although not significant in the use of the Progressive Web Apps (PWA) plugin on the E-Commerce website for PT. Sumber Kurnia Alam. So, it can be concluded that there are six criteria to increase the speed of time in Progressive Web Apps (PWA) based applications namely, first contentful paint, speed index, largest contentful paint, time to interactive, total blocking time, and cumulative layout shift. These six criteria can be improved more quickly if supported by compressing large image and removing unnecessary lines of code. E-Commerce website PT. Sumber Kurnia Alam uses a plugin. Finally, it was decided to use the Super Progressive Web Apps plugin, because when compared to the PWA plugin and PWA for WP & AMP both are not supported for push notifications. Therefore, the Super Progressive Web Apps plugin is the right choice to be implemented on the PT. Sumber Kurnia Alam.

REFERENCES

- [1] N. M. Hui, L. B. Chieng, W. Y. Ting, H. H. Mohamed, and M. R. Hj Mohd Arshad, "Cross-platform mobile applications for android and iOS," *Proc. 2013 6th Jt. IFIP Wirel. Mob. Netw. Conf. WMNC 2013*, pp. 2–5, 2013, doi: 10.1109/WMNC.2013.6548969.
- [2] Y. Makki, "A Comparative study of Android and iPhone Operating System main languages A," *Solid State Technol.*, vol. 63, no. 6, pp. 13651–13658, 2020, [Online]. Available: https://www.researchgate.net/profile/Y-asmin-Makki-4/publication/346956054_A_Comparative_study_of_Android_and_iPhone_Operating_System_main_languages/links/5fd3d48ea6fddccdb8bafda4/A-Comparative-study-of-Android-and-iPhone-Operating-System-main-languages.pdf.
- [3] J. L. Rice, V. V. Phoha, P. Cappelaere, and D. Mandl, "Web farm-inspired computational cluster in the cloud," *Proc. - 2011 3rd IEEE Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2011*, pp. 730–737, 2011, doi: 10.1109/CloudCom.2011.113.
- [4] H. Gillbert Miller, "The spark of innovation begins with collaboration," *Insid. Digit. Ecosyst.*, vol. 11, no. 1, pp. 13–19, 2011.
- [5] M. M. Khan, M. Shams-E-Mofiz, and Z. A. Sharmin, "Development of E-Commerce-Based Online Web Application for COVID-19 Pandemic," *iBusiness*, vol. 12, no. 04, pp. 113–126, 2020, doi: 10.4236/ib.2020.124008.
- [6] P. Que, X. Guo, and M. Zhu, "A Comprehensive Comparison between Hybrid and Native App Paradigms," *Proc. - 2016 8th Int. Conf. Comput. Intell. Commun. Networks, CICN 2016*, pp. 611–614, 2017, doi: 10.1109/CICN.2016.125.
- [7] J. Lee, H. Kim, J. Park, I. Shin, and S. Son, "Pride and prejudice in progressive web apps: Abusing native

- app-like features in Web applications,” *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1731–1746, 2018, doi: 10.1145/3243734.3243867.
- [8] O. Adetunji, C. Ajaegbu, N. Otuneme, and O. J. Omotosho, “Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development,” *Am. Acad. Sci. Res. J. Eng. Technol. Sci.*, vol. 68, no. 1, pp. 85–99, 2020, [Online]. Available: https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/5812.
- [9] A. Gutovets and J. Pustoshilo, “Is PWA the technology of the future?,” 2022.
- [10] N. Pande, A. Somani, S. Prasad Samal, and V. Kakkirala, “Enhanced Web Application and Browsing Performance through Service-Worker Infusion Framework,” *Proc. - 2018 IEEE Int. Conf. Web Serv. ICWS 2018 - Part 2018 IEEE World Congr. Serv.*, pp. 195–202, 2018, doi: 10.1109/ICWS.2018.00032.
- [11] I. Holsby, “The Installation Process of a Progressive Web App Studying the Impact of ‘ Add to Home screen ,” 2021.
- [12] A. Mhaske, A. Bhattad, P. Khamkar, and R. More, “Progressive Web App for Educational System,” *Int. Res. J. Eng. Technol.*, pp. 310–312, 2018, [Online]. Available: <https://arc.applause.com/2015/11/30/application-shell->
- [13] K. Behl and G. Raj, “Architectural Pattern of Progressive Web and Background Synchronization,” *Proc. 2018 Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2018*, no. June, pp. 366–371, 2018, doi: 10.1109/ICACCE.2018.8441701.
- [14] C. Rojas, *Building Progressive Web Applications with Vue.js*. 2020.
- [15] A. Gambhir and G. Raj, “Analysis of Cache in Service Worker and Performance Scoring of Progressive Web Application,” *Proc. 2018 Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2018*, no. June, pp. 294–299, 2018, doi: 10.1109/ICACCE.2018.8441715.
- [16] C. C. B. Bahari and Y. Sumaryana, “Penerapan Progressive Web Apps Pada Aplikasi Lowongan Pekerjaan Dosen Universitas Perjuangan,” *Informatics Digit. Expert*, vol. 1, no. 1, pp. 25–31, 2019, doi: 10.36423/ide.v1i1.285.