JURNAL IPTEK
MEDIA KOMUNIKASI TEKNOLOGI
homepage URL : ejurnal.itats.ac.id/index.php/iptek

Implementation of Hot Code Update in Hybrid Mobile Development Using Dynamic Bundle Loading

Putra Prima Arhandi [1], Ika Kusumaning Putri [2], M. Nintra Zaka [3]

Departement of Informatic Engineering, State Polytechnic of Malang[1,2,3]

ARTICLE INFORMATION

Jurnal IPTEK – Volume 26
Number 2, December 2022

Page:
67 – 74
Date of issue :
December 30, 2022

DOI:
10.31284/j.iptek.2022.v26i2.2372

E-MAIL

[1]putraprima@polinema.ac.id
[2]ikakputri@polinema.ac.id
[3]mnindrazaka@gmail.com

PUBLISHER

LPPM- Institut Teknologi
Adhi Tama Surabaya
Address:
Jl. Arief Rachman Hakim
No.100,Surabaya 60117,
Telp/Fax: 031-5997244

ABSTRACT

Each time a developer wants to release an update to a mobile application, that update has to be reviewed by the marketplace. So the update needs additional time to be delivered to the user. This becomes a problem if that update contains bug fixes that need to be delivered soon to the user. Developers need a solution to deliver updates without waiting the review from the marketplace. Hot code update is the method to add and fix the features of an application without waiting the review from the marketplace. Dynamic bundle loading is one of the methods to implement hot code update in hybrid mobile development. Developer will upload the bundle of javascript source code to a server. Application will download that bundle and apply the feature inside that bundle. In this research the writer will develop a system to implement hot code update using dynamic bundle loading on hybrid mobile development using react native framework. The result of this research shows that hot code updates can release an update to users faster than using google play store with 4 hours 39 minutes 54 seconds time difference. Besides that, the size of update file that must be downloaded is smaller if using hot code update with a difference of 6.1 MB

Keywords: *Hot Code Update, Hybrid Mobile Development, Dynamic Bundle Loading, React Native*

ABSTRAK

Setiap kali developer akan merilis update pada aplikasi mobile, update tersebut akan ditinjau terlebih dahulu oleh marketplace. Sehingga membutuhkan waktu tambahan agar update tersebut bisa sampai ke user. Lamanya proses peninjauan ini bisa menjadi masalah apabila ada perbaikan pada fitur aplikasi yang harus segera diterima oleh user. Sehingga dibutuhkan cara agar user dapat menerima update pada aplikasi mobile tanpa harus menunggu tinjauan dari marketplace. Hot code update merupakan metode yang dapat menambahkan dan memperbaiki fitur pada aplikasi tanpa perlu ditinjau terlebih dahulu oleh marketplace. Dynamic Bundle Loading merupakan salah satu penerapan dari metode hot code update yang dapat diimplementasikan pada hybrid mobile development. Developer akan mengunggah source code javascript terbaru ke server dalam bentuk bundle. Kemudian aplikasi akan mengunduh dan menerapkan fitur - fitur yang ada pada bundle tersebut. Pada penelitian ini akan dibuat sistem untuk menerapkan hot code update menggunakan dynamic bundle loading pada hybrid mobile development menggunakan framework react native. Hasil dari penelitian menunjukkan bahwa metode hot code update dapat merilis update kepada user dengan lebih cepat daripada mengunggah update melalui google play store dengan selisih waktu 4 jam 39 menit 54 detik. Selain itu, ukuran file update yang harus diunduh oleh user lebih kecil apabila menggunakan metode hot code update dengan selisih 6.1 MB

Kata kunci: *Hot Code Update, Hybrid Mobile Development, Dynamic Bundle Loading, React Native*

INTRODUCTION

In this digital era, people's need for smartphones is increasing, this is due to several factors, including the lower prices of smartphones and the wider internet coverage in Indonesia. Based on information from the We are social site, in 2019 the population of Indonesia was 268.2 million, while the number of smartphone users in Indonesia reached 335.5 million. This means that the number of smartphone users is more than the population of Indonesia itself [1].

The high number of smartphone users in Indonesia makes developers compete to create mobile applications that meet the needs of the community. This makes new applications appear on the application marketplace and increases competition among developers. Based on a survey conducted by statista there are already 3.48 million apps available on google play and 2.2 million app available on apple app store [2].

Applications uploaded to the marketplace such as google play or apple app store must meet predetermined criteria, for example, they must not contain viruses or other malicious code that can harm the user. Therefore, the marketplace will review every application that has been uploaded. Based on information obtained from the support.google.com site, the time needed to review applications on the play store can be more than one week [3].

The duration of the application review process in the marketplace can be a problem if there are improvements to the application features that must be available immediately to the user. Based on information from the trentech.id site, Gojek has experienced losses of around tens to hundreds of billions due to a bug in the Go-Pay payment system [4]. If bug fixes on the system are not immediately available to the user, the losses will be even greater. From the problems that have been described, a solution is needed so that users can receive updates on the mobile application without having to wait for reviews from the marketplace.

There are several research proposed to improve the security update for android operating system, the technique involve hot-patching mechanism that will update the android operating system without waiting for the next operating system upgrade [5], [6] and [7]. The hot-patching mechanism for android operating system also can be implemented for the android application. Microsoft has created a system called code push which makes it easier for developers to implement hot code updates in hybrid mobile development. In addition, code push also provides information regarding the log updates that have been released [8]. Another research explore the technique to do a non intrusive update for third party library that used in android app, their research enable app developer to automatically update third party library that has fully backward compatibility [9]. There are also research about hot-patches to fix security issue on android operating system, the research focus on creating a rule driven approach to apply the security fix [10].

Hot code updates are the right method for making minor updates and feature improvements to mobile applications without having to wait for reviews from the marketplace [11]. This can be done because the hot code update method can add and improve features to the application without reinstalling it [12]. There are several research on hot code update, a dynamic bundle loading using several existing plugin on the internet can be used to create a hot code update implementation [13]. However the existing hot code update method are not supporting the regional update feature which is useful for providing updates only in certain regions. An example of the implementation of this regional update is the Instagram application which releases the Instagram music feature only in the United States, Australia, New Zealand, Germany, France, and the United Kingdom due to licensing factors. Another example of using regional updates is to test whether a new feature is developed to meet user needs by releasing it only in certain locations. After the new features are developed according to user feedback, the update can be released globally.

In this research we propose the improved hot code update technique that using a Dynamic Bundle Loading. Dynamic Bundle Loading is an implementation of the hot code update method that can be implemented in hybrid mobile development. Dynamic bundle loading in this research also imroved with regional update feature that will enable hybrid mobile app developer to do an automatic regional update for the android application.

LITERATURE REVIEW

Hybrid Mobile Development

Hybrid Mobile Development is a mobile application development method that combines web technology and the functionality of native code on the associated mobile platform. This strategy allows developers to access the functionality of native code using javascript, for example to access sensors on the device. This can be done by connecting javascript via a bridge with the native programming language on the associated mobile platform. In addition, because it is built using javascript, the resulting application can be released to various mobile platforms using the same javascript source code [11].

Bundling

When developing applications using hybrid mobile development, the developer will build application features in several javascript files. In order for the javascript file to be loaded in the application, the files must be combined into one. So that one javascript file will contain all the features of the application. This merging process is called bundling, and the resulting files are called bundles. Furthermore, if the user runs an application that has been developed, the application will search for and load the bundle file so that the features in the file can be executed [11].

Hot Code Update

Hot code update is a mechanism for updating or improving features in an application without having to re-install it. The purpose of implementing hot code updates is to speed up delivery of updates to users. So that users can complete the update process in a shorter time [11].

Dynamic Bundle Loading

Dynamic bundle loading is a method used in hybrid mobile development to implement hot code updates. First, the developer will write a feature update in the javascript file. Then, the files will be bundled together and uploaded to the server. Then, the application will dynamically check whether any recent bundle files were uploaded to the server or not. If there is, the application will download it and apply the newest bundle. So that the feature updates in the bundle can be used by the user.

METHOD

Hot Code Update Implementation Flow

The method used to apply hot code updates to this system is dynamic bundle loading. The flowchart that describes the process of implementing hot code updates using dynamic bundle loading on the system to be developed are shown in Figure *1*.

System Functionality

The hot code update system has several functionalities for users and developers. The role of the user in this system is to check and download updates through applications that have been installed on the smartphone. Meanwhile, the role of the developer is to develop mobile applications and upload the latest updates through the hot code update information system. The overview of the functionality of the hot code update system are shown in Figure 2.

System Architecture

The explanation of the components in the hot code update system and how they are related to each other are shown in:

1. The frontend of the bundle management information system is used as a developer platform for uploading bundles and releasing new updates.

2. The backend of the bundle management information system is responsible for receiving data and bundle files from the frontend. In addition, the backend is also tasked with receiving update check requests from the dynamic bundle loading plugin and providing the latest updates when new updates are found.
3. The bundle management information system database is responsible for storing data received by the backend, for example developer account data, project data, and update logs.
4. Cloudinary is responsible for storing the bundle files sent by the frontend through the backend whenever the developer releases a new update.

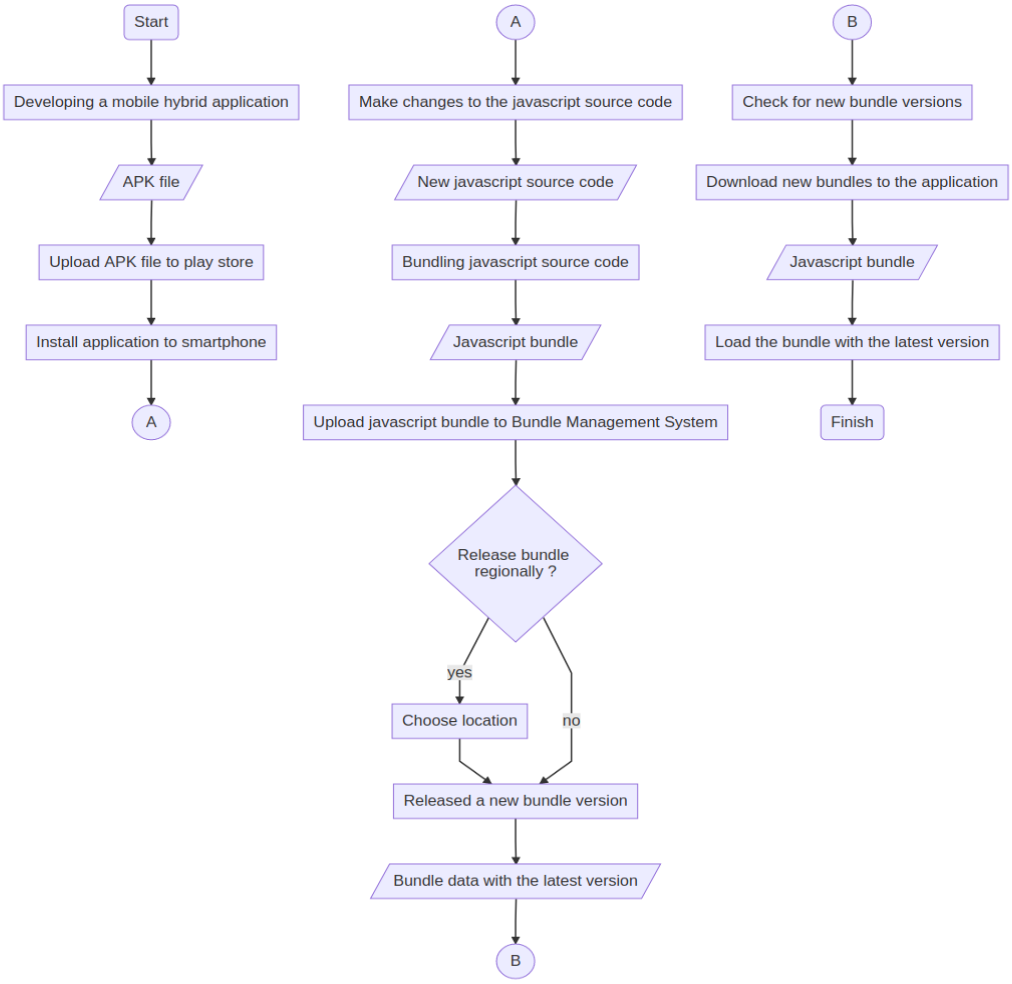
Start
Developing a mobile hybrid application
APK file
Upload APK file to play store
Install application to smartphone
A
A
Make changes to the javascript source code
New javascript source code
Bundling javascript source code
Javascript bundle
Upload javascript bundle to Bundle Management System
Release bundle regionally ?
yes
no
Choose location
Released a new bundle version
Bundle data with the latest version
B
B
Check for new bundle versions
Download new bundles to the application
Javascript bundle
Load the bundle with the latest version
Finish

Figure 1. Process Implementation Of Hot Code Update Using Dynamic Bundle Loading

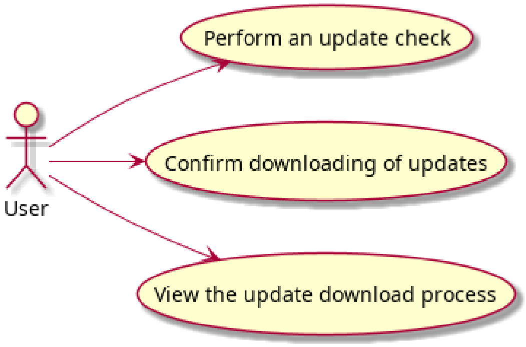
User
Perform an update check
Confirm downloading of updates
View the update download process

Figure 2. The overview of the functionality of the hot code update system

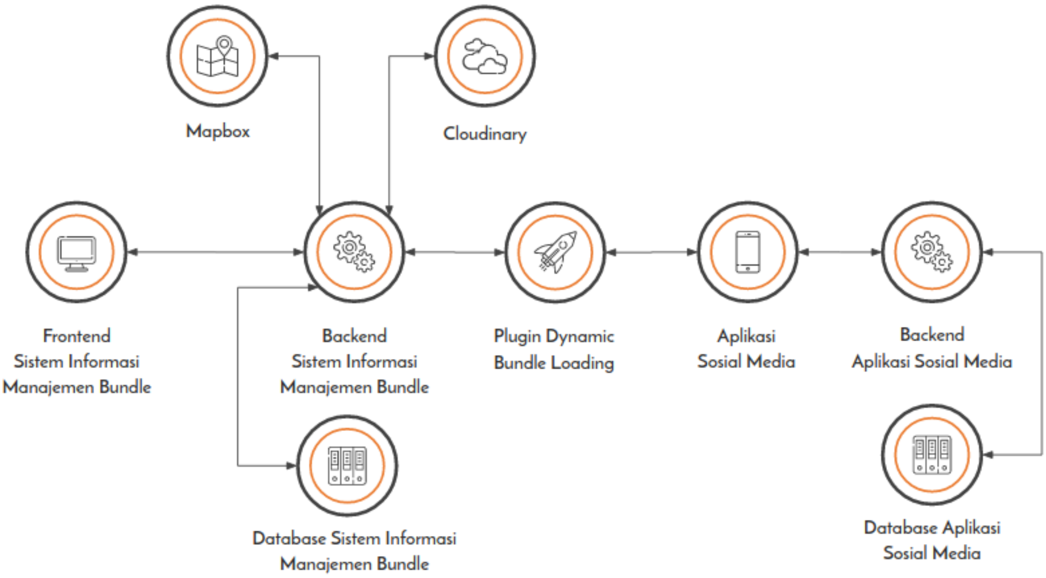
Mapbox
Cloudinary
Frontend Sistem Informasi Manajemen Bundle
Backend Sistem Informasi Manajemen Bundle
Plugin Dynamic Bundle Loading
Aplikasi Sosial Media
Backend Aplikasi Sosial Media
Database Sistem Informasi Manajemen Bundle
Database Aplikasi Sosial Media

Figure 3. Relation of component in hot code update system

5. To perform a regional update, the backend of the bundle management information system will store the coordinates of the location to be applied in the form of latitude and longitude. The coordinates of implementing the update will be compared with the coordinates of the user. Before being compared, the coordinates will be converted into a province name. So that users will receive updates if they are in the same province as the province where the update is implemented. Mapbox is responsible for changing the name of the province to latitude and longitude (forward geocoding) when the backend will store the location of the update implementation. Mapbox is also tasked with changing latitude and longitude into province names (reverse geocoding) when the backend will compare the user's location and the location of the update implementation.
6. The dynamic bundle loading plugin is a library installed by the developer in the application being developed. This plugin is in charge of sending requests and user locations to the backend of the bundle management information system to check for updates. When a new update is received by the backend, this plugin will save the bundle of that update to internal storage and apply it when the app is reopened.
7. Social media application is an application that is used as an example of the application of hot code updates in this thesis. Users can create accounts, view profiles, and create new posts.
8. The backend of the social media application is responsible for receiving data sent by the social media application, for example user account and post data.
9. The social media application database is responsible for storing data sent by the application via the backend.

RESULT AND DISCUSSION

Update Time Comparison

In order to compare the duration of implementing updates using hot code updates compared to uploading updates to the Google Play Store, a social media application has been developed and several additional features as shown in the table below. The start time is calculated from the time the developer starts uploading the update. Meanwhile, the finished time is calculated after the update can be downloaded by the user.

Table 1. Comparison Update Time Between Play Store and Hot Code Update

Feature Name	Start Time		Finish Time		Total	
	PS	HCU	PS	HCU	PS	HCU
Password Confirmation	23-06-2020 19:47	27-06-2020 14:00:00	24-06-2020 00:00	27-06-2020 14:01:28	4h 13m	1m 28s
View Other Users Profile	24-06-2020 18:00	27-06-2020 15:00:00	25-06-2020 00:00	27-06-2020 15:00:55	6h	55s
Display User Post on The Profile Page	25-06-2020 20:10	27-06-2020 16:00:00	26-06-2020 00:00	27-06-2020 16:00:55	3h 50m	55s
Average					4h 41m	1m 6s

Note : PS (Google Play Store), HCU (Hot Code Update)

Based on the results from the Table 1, it can be seen that the hot code update method using dynamic bundle loading takes an average of 1 minute 6 seconds for the update to be downloaded by the user. Meanwhile, if the update is uploaded to the Google Play Store, it takes an average of 4 hours 41 minutes for the update to be downloaded by the user. This occurs because it takes additional time to review uploaded updates. Plus, updates that have been reviewed often have a delay before the update can be seen on the Google Play Store. So it can be concluded that the hot code update method can release updates to users more quickly with a time difference of 4 hours 39 minutes 54 seconds.

File Size Comparison

The following is a comparison of the file sizes that users must download using hot code updates using the Google Play Store which can be seen in the table below.

Table 2. Update File Size Comparison Beetween Play Store and Hot Code Update

Feature Name	File Size	
	PS	HCU
Password confirmation	8,4 Mb	2,3 Mb (2.373.373 Bytes)
View others users' profile	8,4 Mb	2,3 Mb (2.373.674 Bytes)
Displays user posts on the profile page	8,4 Mb	2,3 Mb (2.373.880 Bytes)
Average	8,4 Mb	2,3 Mb (2.373.642,33 Bytes)

Note : PS (Google Play Store), HCU (Hot Code Update)

Based on the results from the Table 2, it can be seen that if the developer uploads an update to the Google Play Store, the average file size that the user must download every time there is an update is 8.4 MB. Meanwhile, if the developer uploads the update using hot code update, the average file size that must be downloaded by the user every time there is an update is 2.3 MB. So it can be concluded that by using hot code update, the file size that must be downloaded by the user will be smaller with a difference of 6.1 MB. This can happen because if you download an update via the Google Play Store, the user must download the entire APK file. Meanwhile, if you download it using hot code update, all you need to download is the javascript bundle.

CONCLUSIONS

Based on the results of the experiment above, we conclude that hot code update using dynamic bundle loading can be used as an alternative for hybrid mobile app developers to release an update to their mobile app in Google Play Store. The update process using hot code update are superior to Google Play update in terms of update time and file update size.

REFERENCES

[1] “Digital 2020: Indonesia,” *DataReportal – Global Digital Insights*. https://datareportal.com/reports/digital-2020-indonesia (accessed Nov. 23, 2021).

[2] “Biggest app stores in the world 2020,” *Statista*. https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/ (accessed Nov. 23, 2021).

[3] “Publish your app - Play Console Help.” https://support.google.com/googleplay/android-developer/answer/9859751?hl=en (accessed Nov. 23, 2021).

[4] “Gojek Rugi Puluhan Milyar Gara-gara Bug GPC | Trentech.id.” https://www.trentech.id/gojek-rugi-puluhan-milyar-gara-gara-bug-gpc/ (accessed Nov. 23, 2021).

[5] Y. Chen, Y. Zhang, Z. Wang, T. Wei, and L. Xia, “Adaptive Android Kernel Live Patching,” p. 19.

[6] G. Russello, A. B. Jimenez, H. Naderi, and W. van der Mark, “FireDroid: hardening security in almost-stock Android,” in *Proceedings of the 29th Annual Computer Security Applications Conference*, New Orleans Louisiana USA, Dec. 2013, pp. 319–328. doi: 10.1145/2523649.2523678.

[7] C. Mulliner, J. Oberheide, W. Robertson, and E. Kirda, “PatchDroid: scalable third-party security patches for Android devices,” in *Proceedings of the 29th Annual Computer Security Applications Conference*, New Orleans Louisiana USA, Dec. 2013, pp. 259–268. doi: 10.1145/2523649.2523679.

[8] lucen-ms, “Use CodePush to update your app live - Visual Studio App Center.” https://docs.microsoft.com/en-us/appcenter/distribution/codepush/ (accessed Nov. 23, 2021).

[9] Y. Duan, L. Gao, J. Hu, and H. Yin, “Automatic Generation of Non-intrusive Updates for Third-Party Libraries in Android Applications,” p. 16.

[10] Y. Chen *et al.*, “InstaGuard: Instantly Deployable Hot-patches for Vulnerable System Programs on Android,” presented at the Network and Distributed System Security Symposium, San Diego, CA, 2018. doi: 10.14722/ndss.2018.23124.

[11] W. Liu, “Discussion on Hot Update Mechanism of Mobile Application,” vol. 86, p. 7.

[12] W. Danielsson, “React Native application development,” p. 70.

[13] C. I. Lemes, M. Willocx, V. Naessens, and M. Vieira, “An Analysis of Mobile Application Update Strategies via Cordova,” p. 8, 2017.

This Page Intentionally Left Blank