



# **Implementation of Google Apps and Mit App Inventor on Android Based Real Time Attendance Report Application**

Oktoverano Hendrik Lengkong  
Universitas Klabat, Faculty of Computer Science  
*Oktoverano@unklab.ac.id*

## **ABSTRACT**

Attendance in educational institutions is an important thing that is always a common concern, attendance is used to measure student performance or seriousness in taking education. Klabat University as an Educational institution, in this case at the Faculty of Computer Science. Trying to manage attendance as well as possible to get information that is fast, precise and accurate. with this information about attendance, this institution can find out what things need to be improved and what things should be limited, or what decisions must be made for student success, which are related to attendance. The need for attendance reports at the Computer Faculty of Klabat University is urgently needed. Currently attendance report is still using paper, and the process for creating the report are by manual input one by one for generating the report. This research will create an application that runs on the Android operating system and this application will later be accessed with an Android-based smartphone with a student ID card barcode scan feature, as a marker of student attendance, so the report can be generated automatically into the system. The research model or method used is the Prototyping method. Prototyping design models are made faster so that the time needed for system development is shorter. This characteristic is very suitable with the needs of the existing system that only requires a short time in its development. The tools that will be used by researcher are MIT App Inventor to build an Android application. For the purposes of real time reports of student attendance, researcher will use Google apps as a data storage facility and as a means for generating reports. The results of this study are expected to create an android-based mobile application that in real time can send student attendance data that can be accessed directly anytime and anywhere in the form of attendance report data. For future development, researcher may suggest using other method than barcode scanning. RFID or NFC tag supported for student ID Card can be used for attendance check.

**Keywords:** Attendance Report, Android Application, ID Card, Prototyping Model, MIT App Inventor, Google Apps.

## **INTRODUCTION**

In teaching and learning activities, attendance reports are mandatory. Based on these data reports, it can be used as a means of information needed to be one of the performance measurements or provide an assessment of students who are currently studying.

Ways to collect attendance data and make Attendance reports vary. In this research, case studies were taken at the Klabat University (Unklab) campus, Airmadidi, North Sulawesi. Especially at the Faculty of Computer Science. As for how to take the presentation as follows: Each public lecture meeting, starting in the morning at 08.30. The time span given to students is ten minutes to get ready to arrive in the room, calculated from the time it starts. If students arrive in the room after the 10 minutes they are given, students will be declared late. If students come past 20 minutes from the time the public lecture starts, the student will be declared absent or absent. Retrieval of attendance data is carried out by the officer whose Attendance is designated, in this case usually referred to as an absenter. Absenter will walk one by one to the seat that has been determined. And the data collection is done by calling the name and writing it on a sheet of paper and asking each student to sign a report on attendance.

After the attendance data is collected by the absenter, a report will be made. With procedure; each paper from each absenter will be collected and put together, and then will be taken to the office of the computer science faculty to be recorded one by one made by the faculty administration officer.

Through the process of data collection and reporting of this Attendance, what is needed from both students and the faculty of computer science (FIK) is the process of taking and reporting that is fast and accurate. But based on the previous explanation, the needs are still far to be fulfilled. With this need, researchers will create an application that is expected to be able to meet the need for attendance report that are fast and accurate or in real time terms using google docs.(Attebury, George, Judd, & Marcum, 2013)

Technology has an important role especially in Educational activities.(Ventayen & Orlanda-Ventayen, 2018) The application will be developed based on Android.(Maia, Nogueira, & Pinho, 2010) Data shows that currently smartphone users based on Android are quite high.(Jkielty, 2019) Therefore, in this study the researchers chose to develop applications with the Android operating system. The application will be built using Google Apps Script and MIT App Inventor.(MIT, 2019) Apps Script is a scripting language for light-weight application development in the G Suite platform.(Google Developer, 2018) For media recording or reporting data will use Cloud Computing through Google docs: spreadsheet.(Spaeth & Black, 2012) The Google Script that will build the script to connect the data to be input, and MIT App Inventor will create an android-based application interface.

## **LITERATURE REVIEW**

In this research, the Google Apps platform is used as a means to create reports, namely through one part of the application, namely Google Sheets. Attendance reports will be displayed through a template that has been created. In order to display information or Attendance data in real time, a part of Google Apps will be used, the Google Script application.

Google Apps is a series of web-based applications from Google consisting of communication and productivity tools. There are various editions of the Google Apps suite, which are targeted at groups such as businesses, schools, and other organizations. Google Apps includes the following applications: Google Mail (also known as Gmail), Google Calendar, Google Talk, Google Docs, Google Sites, and Google Video. Each of these applications is entirely web-based, although there are client programs that provide additional functionality. Because it is web-based, every Google Application is inherently cross platform; if the computing platform supports modern web browsers, it has full compatibility without requiring additional software. Basically Google Apps is a web-based model, can be scaled, and basically virtualized as an example of modern cloud computing. (Herrick, 2009)

Google Apps Script is a fast application development platform that makes it fast and easy to create business applications that are integrated with G Suite. Write code in JavaScript and have access to the default library for favorite G Suite applications such as Gmail, Calendar, Drive and more. There is nothing to install because there is a code editor right in the browser, and the script runs on Google's servers. (Google Developer, 2018)

Spreadsheets are designed to meet the needs of dynamic organizations. With the AI feature, you can get the right analysis to make significant business decisions. Cloud-based architecture allows you to collaborate with anyone, anytime and anywhere. Compatibility with various external systems, including Microsoft Office, minimizes the problems that commonly occur when working with various data sources. Thanks to Google's infrastructure, Sheets can give you freedom to be creative, while maintaining the security of your information. (Google, 2019)

MIT App Inventor is an intuitive visual programming that allows everyone from children to adults to build fully functional applications for smartphones and tablets. Those who are new to MIT App Inventor can have a simple first application that runs in less than 30 minutes. And what's more, this block-based tool facilitates the creation of complex and high-impact applications in a much shorter time than traditional programming environments. The MIT App Inventor project seeks to democratize software development by empowering everyone, especially young people, to move from technology consumption to technology creation. This

visual programming framework is related to MIT's scratch programming language, which specifically is an implementation of the Open Block distributed by the MIT Scheller Teacher Education Program taken from research conducted by Ricarose Roque. App Inventor uses the Kawa Language Framework and Kawa's dialect - which was developed by Per Bothner and distributed as part of the GNU Operating System by the Free Software Foundation as a Compiler which translates visual block programming to be implemented on the Android platform. (Adi, 2011)

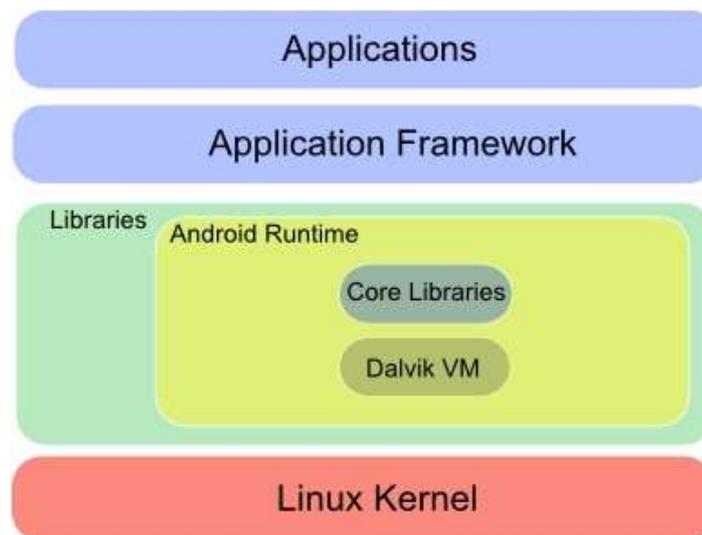


Figure 1. **Android Structure**

Android is an open source software architecture provided by the Open Handset Alliance, the Android Platform includes an operating system, middleware, and applications. As for features, Android combines common features found today on any mobile device platform, such as: reuse of the application framework, integrated browser, optimized graphics, media support, network technology, and other features. The Android architecture, illustrated in Figure 6, consists of five layers: Application, Application Framework, Library, Android until then and finally the Linux kernel. The top layer, the Application layer, provides a core set of applications that are generally offered outside the box with any mobile device. The Application Framework layer provides the Application Programming Interfaces (APIs) framework that is used by applications running at the very top layer. In addition to the API, there is a set of services that allow access to Android's core features such as graphical components, information exchange managers, event managers and activity managers, Android provides its own VM, as stated earlier, named Dalvik. Dalvik was designed from the start and is specifically targeted for

memory and CPU-restricted devices. It runs Java applications on it and unlike standard Java VMs, which are stack based, Dalvik is an unlimited register based machine. Being a register machine, it presents two advantages compared to stack-based machines. That is, it requires 30% fewer instructions to do the same calculation as an ordinary stack machine, causing a reduction in instruction delivery and memory access; and less computational time, which also comes from removing common expressions from instructions. However, Dalvik represents 35% more bytes in the instruction stream than an ordinary stack machine. This deficiency is compensated by the consumption of two bytes when consuming instructions. Dalvik uses its own byte-code format name, Dalvik Executable (.dex), with the ability to include several classes in one file. It is also able to do some optimizations during dex generation when it comes to internal storage of types and constants by using principles such as minimal repetition; per-type ponds; and implicit labeling. By applying these principles, it is possible to have a smaller dex file than a typical Java archive file (jar). During installation time, each dex file is verified and optimized such as byteswapping and padding, static-linking and in-lining methods are carried out to minimize runtime evaluation and at the same time to avoid code security breaches. The Linux kernel, version 2.6, is the lowest layer and is also a hardware abstraction layer that allows top layer interaction with the hardware layer through device drivers. In addition, it also provides the most basic system services such as security, memory management, process management and network stacks. (Maia et al., 2010)

## **METHODS**

In this study, the method used is a software engineering research. Research that applies science into a design in order to obtain performance in accordance with specified requirements. The design is a synthesis of design elements combined with scientific methods into a model that meets certain specifications. The research is directed to prove that the design meets the specified specifications. The research begins by determining the design specifications that meet the specified specifications, choosing the best alternative, and proving that the design chosen can meet the specified requirements in an efficient, effective and relatively inexpensive way. (Sommerville, 2010)

### **Data collection procedure:**

In this study, the data collection procedure was carried out by means of documentation. Where application development is carried out using the internet network with the website as a tool to build programming code structures.

### **Data collection technique:**

#### 1. Literature

In this study, researchers conducted a literature study, from sources: Books, Journals, Research, websites.

#### 2. Experiments

Researchers conduct direct experiments on research objects using an Android-based smartphone.

### **Research Instrument:**

#### 1. Hardware

- a. Android Based Smartphone
- b. Computer with minimum requirements:
  - Windows 7, 8.1
  - RAM Memory 4GB
  - Processor Quad Core i3 3.0 GHz
  - HDD 100 GB free space
- c. Wireless router

or higher

#### 2. Software

- a. MIT App Inventor
- b. Adobe Photoshop CS6
- c. Microsoft Office 2016
- d. Google Chrome Browser

## **RESULTS**

The application was developed using MIT App Inventor, which was developed using scratch visual programming framework from MIT. An implementation of the Open Block distributed by the MIT Scheller Teacher Education Program. The programming language uses Google Apps Script with the concept of Cloud Computing, which is stored in a Google Account.



Figure 3. Main Interface MIT App Inventor.

Figure 3 displays the interface of the MIT App Inventor. Which has a function to create an interface from this Attendance application. Through this research researchers found the use of the MIT App Inventor is able to create an interface that can be used to create a Attendance application. Its features are quite complete. MIT App Inventor is also easier in terms of application testing. For testing you can use the default emulator that is provided, or with other options, namely using the emulator installed on an Android-based smartphone. Using the Emulator is also quite easy just by scanning a barcode or entering the code displayed to be entered into an Android-based emulator application from MIT App Inventor which can be downloaded at Google Play store.

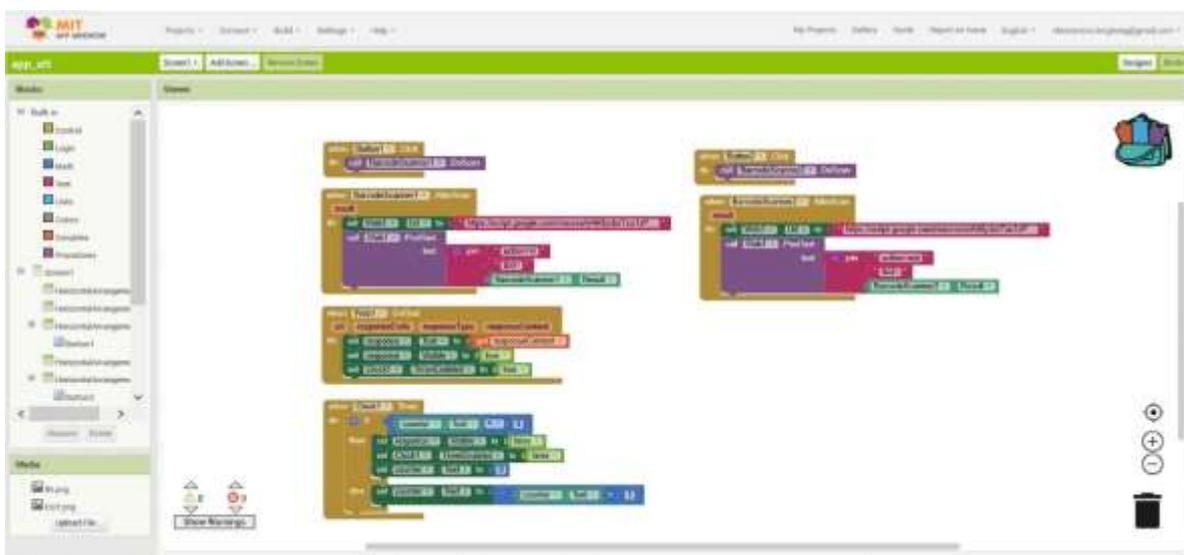
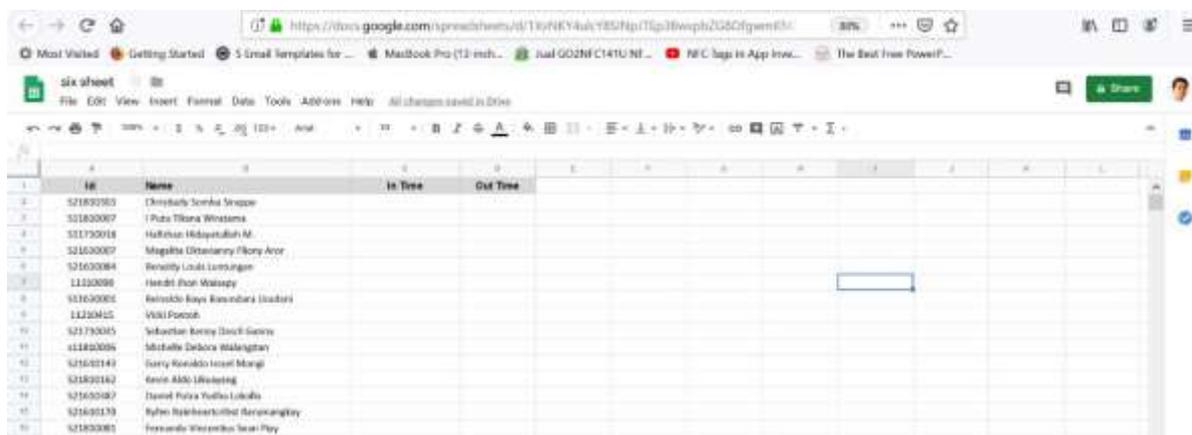


Figure 4. Block Programming Interface MIT App Inventor.

In Figure 4, we can see the concept of block programming in the MIT App Inventor. Although impressed relaxed, but the Block Programming provided by MIT App Inventor is quite complete. And in this case is able to answer the need for the Attendance application in this research. The concept of Block Programming is complete with functions such as: control, logic, math, text, lists, colors, variables, procedures.



	ID	Name	In Time	Out Time
1				
2	52182783	Christy Sornha Srepper		
3	52182007	I Putu Tilona Winatama		
4	52173018	Hafidhan Hidayatullah M.		
5	52163007	Megalia Ditanary Flory Anor		
6	52163084	Rendy Luthi Lumbigan		
7	1123008	Hendri Fion Walagay		
8	52163001	Reincho Rasy Ramadhani Usahari		
9	1123045	Widi Poozoh		
10	52173045	Selmahan Keny Dauli Setiyo		
11	52182006	Miftaha Debora Walangtan		
12	52163243	Geary Kevinda Inesl Mungil		
13	52182182	Kevin Aldo Wibisono		
14	52163087	Daniel Putra Yudha Lokohi		
15	52163233	Nylen Rajaherathilal Ranawongkay		
16	52182001	Fernando Winandha Suan Pity		

Figure 5. Google Docs Spreadsheet Attendance Report.

In Figure 5, we can see the data in the form of Attendance reports that are displayed in realtime. Where this data is obtained using an application created by researchers using the tool from the MIT App inventor described earlier in Figures 2 and 3. Through the Android application interface, the smartphone uses a barcode scanner to scan the barcodes available on the student ID cards. After scanning if the data received matches the existing data in the student database stored in the google docs spreadsheet then the check in time data will be entered in the check in time data field, as well as the check out procedure the same way applies. Check in and check out are only distinguished by the menu choices.



Figure 6. Display interface Attendance applications based on Android.

Figure 6 is a display on the Attendance application interface. There are two menus there, namely IN and OUT. Users can choose IN if they want to input attendance data, then attendance data in the form of check-in time will be recorded in a database based on the scan results of each student ID number. If students have finished attending classes in general classes, students must check out using the same procedure. As one of the objectives of this research is to want attendance data in real time and accurately. By using the MIT App Inventor and Google Apps Script and Google Docs Applications can be built and produce applications that match what is needed.

## DISCUSSION

### The Conceptual Framework

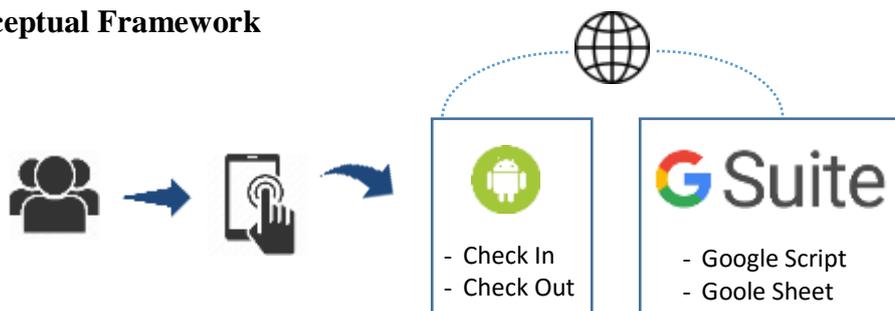


Figure 2. Conceptual Framework of Apps.

The explanation in Figure 2 as the conceptual framework of the application is as follows:

1. The user activates a smartphone with an android operating system
2. The user activates the Attendance report application:
  - a. The user selects the check in option
  - b. The user selects the check out option

- c. The application adds Attendance data
3. Applications through the internet network add data into google apps, based on data entered through the application.
4. The application displays notification of check-in and check-out time.

### **Scope and limitations**

The following is the scope of the research made:

1. Use a Barcode reader to read student ID Card data
2. The data added shows the time of entry and exit time
3. Real time report data.

The limitations in this study are:

1. The Attendance report application requires an internet network as a means for real time reporting.
2. This application does not automatically calculate time for students who are late
3. The application only displays attendance status and does not count attendance.
4. The application only displays daily real time reports.

### **Conclusion**

MIT App Inventor, running well at the time of implementation. Can produce android applications that can be used in an Android Smartphone. Google Apps Script can give commands directly into Google Docs Spreadsheet files. The .Apk extension file can be directly downloaded via the MIT App Inventor and can be installed on an Android smartphone.

Applications can still be maximized, for that suggestion from the author, for further development can be added features such as; login function to find out who the user is in charge of scanning, function to add profile pictures to the database. Because it still uses Barcodes, for the following developments it can use RFID or NFC Cards to avoid barcode scanning methods. This is better, because there are times when the barcode is difficult to read because the condition of the barcode printing on the card starts to fade or is erased, which results in the difficulty of the scanning process.

## REFERENCES

- Adi. (2011). Android Indonesia. Retrieved February 4, 2019, from <http://indo-android.blogspot.com/2011/09/apa-itu-app-inventor.html>
- Attebury, R., George, J., Judd, C., & Marcum, B. (2013). Google Docs: A Review. *Against the Grain*. <https://doi.org/10.7771/2380-176x.2736>
- Google. (2019). Google Sheets. Retrieved February 4, 2019, from [https://gsuite.google.com/intl/id/products/sheets/?utm\\_source=google&utm\\_medium=cp&utm\\_campaign=japac-ID-all-id-dr-bkws-all-golden-trial-b-dr-1003989&utm\\_content=text-ad-none-none-DEV\\_c-CRE\\_264189258409-ADGP\\_Hybrid%257CAWSEM%257CBKWS~BMM%257CSheets%257](https://gsuite.google.com/intl/id/products/sheets/?utm_source=google&utm_medium=cp&utm_campaign=japac-ID-all-id-dr-bkws-all-golden-trial-b-dr-1003989&utm_content=text-ad-none-none-DEV_c-CRE_264189258409-ADGP_Hybrid%257CAWSEM%257CBKWS~BMM%257CSheets%257)
- Google Developer. (2018). Google Apps Script. Retrieved February 4, 2019, from <https://developers.google.com/apps-script/overview>
- Herrick, D. R. (2009). Google this! Using Google apps for collaboration and productivity. In *SIGUCCS'09: proceedings*. <https://doi.org/10.1145/1629501.1629513>
- Jkielty. (2019). Android v iOS market share 2019. Retrieved September 30, 2019, from <https://deviceatlas.com/blog/android-v-ios-market-share>
- Maia, C., Nogueira, L., & Pinho, L. (2010). Evaluating android os for embedded real-time systems. *Published in Proceedings of the 6th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications, Brussels, Belgium*. <https://doi.org/10.1089/lap.2006.05083>
- MIT. (2019). MIT App Inventor | Explore MIT App Inventor.
- Sommerville, I. (2010). Software Engineering: Principles and Practice. *Software Engineering Journal*. <https://doi.org/10.1049/sej.1994.0029>
- Spaeth, A. D., & Black, R. S. (2012). Google docs as a form of collaborative learning. *Journal of Chemical Education*. <https://doi.org/10.1021/ed200708p>
- Ventayen, R. J. M., & Orlanda-Ventayen, C. C. (2018). Google's Gsuite Applications in Open University System's Perspective. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3174082>