

# Laporan Kehadiran secara *Real-Time* Pada Kuliah Umum berbasis *Mobile Apps* menggunakan *MIT App Inventor* (Studi Kasus : Fakultas Ilmu Komputer – Universitas Klabat)

## *Real-Time Attendance Report on General Class based on Mobile Apps using the MIT App Inventor (Case Study: Faculty of Computer Science – Universitas Klabat)*

Oktoverano Lengkong<sup>1</sup>, Oktovianus Masrikat<sup>2</sup>

<sup>1,2</sup>Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Klabat,  
Jl. A. Mononutu, Airmadidi – Minahasa Utara, Sulawesi Utara  
e-mail: [1oktoverano@unklab.ac.id](mailto:oktoverano@unklab.ac.id), [2s11520005@student.unklab.ac.id](mailto:s11520005@student.unklab.ac.id)

### ABSTRAK

Laporan kehadiran *real-time* Kuliah Umum Fakultas Ilmu Komputer – Universitas Klabat menggunakan teknologi mit app inventor merupakan aplikasi pencatatan dan pelaporan absen kegiatan kuliah umum secara cepat dengan tingkat keakuratan yang tepat. Kehadiran merupakan suatu rutinitas dan bagian penting dalam setiap instansi pendidikan untuk membuktikan keaktifan dan keseriusan mahasiswa dalam perkuliahan. Faktor kehadiran pada kegiatan *Kuliah Umum* menjadi salah satu tolok ukur bagi seorang mahasiswa untuk dapat melanjutkan perkuliahan pada semester berikutnya. Adalah suatu keperluan untuk membangun suatu sistem pelaporan dan pencatatan kehadiran bagi kegiatan *Kuliah Umum* bagi Fakultas Ilmu Komputer yang masih melakukan pencatatan secara manual yaitu dengan menjalankan formulir absen dalam bentuk kertas dan butuh waktu yang lama terkait pencatatan atau pun paraf kehadiran. Penelitian dengan menggunakan implementasi dari mit app inventor ini dibuat untuk membangun sebuah aplikasi yang akan dijalankan pada *smartphone* dengan sistem operasi android yang mendukung fitur pemindaian *qr code/barcode* pada *ID card* mahasiswa untuk menghasilkan laporan secara cepat dan otomatis ke dalam database menggunakan fitur *scan in time* dan *scan out time* serta fitur *delete record* untuk menghapus seluruh data yang telah di *record*. Adapun metode yang digunakan dalam penelitian ini yaitu dengan menggunakan metode *prototype* yang dibuat lebih cepat sehingga dalam pengembangan membutuhkan waktu yang singkat. Dalam penelitian ini peneliti menggunakan media *spreadsheet* berbasis *cloud* sebagai database dan juga *app script* sebagai sarana untuk membangun program yang dapat menjalankan fungsi *blockcodes* dari *mit app inventor* untuk pencatatan dan pelaporan absen *Kuliah Umum* secara *real-time* dan secara langsung dapat diakses oleh pihak fakultas kapanpun dan dimanapun. Pengujian aplikasi ini telah dilakukan pengujian pada *smartphone* Android dan berhasil di pasang pada Android versi 5.1.1 ke atas.

**Kata Kunci:** Laporan Absen, Aplikasi Android, MIT App Inventor, *ID Card*, *Scan In Time*, *Scan Out Time*, *Delete Record*.

### ABSTRACT

*The real-time attendance report of the General Class Faculty of Computer Science - University of Klabat uses mit app inventor technology to record and report attendance for public lectures quickly with the right level of accuracy. Attendance is a routine and an important part in every educational institution to prove the activeness and seriousness of*

students in lectures. The attendance factor in general class activities is one of the benchmarks for a student to be able to continue his studies in the next semester. It is necessary to build a system for reporting and recording attendance for general class activities for the Faculty of Computer Science who are still doing manual records, namely by running the absences form in paper and it takes a long time to record or initial attendance. Research using the implementation of the mit app inventor was made to build an application that will run on a smartphone with an Android operating system that supports the QR code / barcode scanning feature on student ID cards to generate reports quickly and automatically into the database using the scan in time feature and scan out time and delete record feature to delete all recorded data. The method used in this research is by using the prototype method which is made faster so that development requires a short time. In this study, researchers used cloud-based spreadsheet media as a database and also an app script as a means to build a program that can run the block code function of the mit app inventor for recording and reporting general class absences in real-time and directly accessible to the faculty anytime and anywhere. Testing of this application has been tested on an Android smartphone and was successfully installed on Android version 5.1.1 and above.

**Keywords:** *Absence Report, Android Application, MIT App Inventor, ID Card, Scan In Time, Scan Out Time, Delete Record.*

## 1. PENDAHULUAN

Universitas Klabat (Unklab) merupakan salah satu universitas swasta di provinsi Sulawesi Utara yang berada dibawah naungan Gereja Masehi Advent Hari Ketujuh (GMAHK) dengan memiliki 7 konsentrasi bidang keilmuan diantaranya Sekretaris, Ekonomi, Komputer, Keperawatan, Pendidikan, Filsafat Agama Kristen, serta Pertanian[2]. Seiring perkembangan zaman yang begitu pesat dan semakin modern khususnya dalam bidang teknologi informasi, dibutuhkan adanya suatu sistem informasi. Sistem informasi sangat bermanfaat dan berguna dalam memenuhi kebutuhan seseorang dan mempermudah menyelesaikan suatu tugas dengan baik dan cepat [3].

Adapun bukti nyata untuk memenuhi tuntutan perkembangan tersebut yaitu adanya peningkatan dalam berbagai fasilitas pendukung salah satunya adalah Sistem Informasi Unklab (SIU). SIU sangat bermanfaat dalam menunjang kegiatan akademis dan juga bermanfaat dalam segi penyediaan informasi diantaranya pada saat proses pendaftaran, penyusunan jadwal, kontrak mata kuliah dan informasi dari nilai tiap mata kuliah yang sudah pernah dikontrak dan juga menyimpan informasi nama dari setiap mahasiswa yang sudah terdaftar pada setiap fakultas [4].

Aktivitas perkuliahan dapat berlangsung dengan baik apabila mahasiswa turut aktif untuk menghadiri setiap pertemuan kuliah umum yaitu kegiatan *Kuliah Umum* yang diselenggarakan sekali dalam seminggu oleh setiap fakultas. Karena itu dibutuhkan suatu sistem pengawasan yang baik menyangkut kehadiran mahasiswa yaitu kehadiran [5]. Kehadiran merupakan suatu hal yang sangat penting dan tidak bisa terlepas dari suatu pertemuan kelas ataupun *Kuliah Umum*. Sistem pencatatan dan pelaporan absen diperlukan sebagai bukti bahwa seorang mahasiswa hadir atau tidak pada suatu pertemuan kelas maupaun *Kuliah Umum* [6].

Adapun beberapa ketentuan mengenai absen sesuai dengan peraturan yang berlaku di Unklab. Seseorang dapat dikatakan absen atau alpa jika yang bersangkutan tidak hadir secara fisik di suatu pertemuan baik itu pertemuan kelas ataupun *Kuliah Umum*. Jika seseorang kedapatan masuk *Kuliah Umum* sepuluh menit setelah bel ketiga atau *late* sebanyak tiga kali maka akan dihitung sebagai satu absen. Jika seseorang berhalangan hadir dengan berbagai alasan yang patut maka akan tetap terhitung absen.

Dalam aktivitas perkuliahan saat ini proses pencatatan dan pelaporan absen dalam kegiatan *Kuliah Umum* masih menggunakan cara manual. Setiap fakultas memiliki cara

tersendiri dalam melakukan pencatatan absen. Ada yang memilih mencatat kehadiran dengan cara menjalankan *form* absen kemudian diparaf, ada juga yang mencatat kehadiran dengan melihat *seating arrangement* yang telah diatur dan disepakati bersama sehingga dengan demikian dapat membutuhkan waktu yang lama pada saat pencatatan absen terlebih jika mahasiswa yang terdaftar pada satu fakultas memiliki jumlah yang banyak. Adapun media pencatatan absen yang secara fisik masih dalam bentuk kertas sehingga tingkat keamanan laporan dokumentasi dapat terbilang lemah, dalam hal ini media pencatatan absen dapat hilang, rusak maupun tercecer sebelum diinput oleh pihak fakultas.

Adapun penelitian terkait mengenai sistem kehadiran yang digunakan sebagai salah satu pedoman dalam penulisan ini yaitu *Implementation of Google Apps and Mit App Inventor on Android Based Real Time Attendance Report Application*[7], yang mana pada penelitian ini juga menggunakan *MIT App Inventor* untuk membangun aplikasi *android* yang bertujuan untuk membuat laporan kehadiran siswa secara *real-time* pada Fakultas Ilmu Komputer – Universitas Klabat, dengan menggunakan aplikasi *google drive* sebagai fasilitas penyimpanan data pada *cloud* dan sebagai sarana untuk menghasilkan laporan dengan sistem *scan QRcode*. Dalam Penelitian ini peneliti melakukan pengembangan dari penelitian tersebut, namun pada penelitian kali ini peneliti sudah menambahkan fungsi untuk *men-generate late* dan absen dari waktu yang didapatkan dan menerapkan sistem ini pada kegiatan *Kuliah Umum*.

Berdasarkan permasalahan yang telah diuraikan maka peneliti ingin merancang dan membuat suatu aplikasi Laporan Kehadiran secara *real-time* dengan memanfaatkan perangkat lunak *Mit App Inventor* yang dapat mengatasi beberapa masalah dalam sistem kehadiran yang masih dilakukan dengan pelaporan presensi tanpa bantuan teknologi.[1]

## 2. METODE PENELITIAN

### 2.1 Android

Android adalah sistem operasi untuk *smartphone* yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia* [8].

### 2.2 Qrcode

*QRCode (quick response)* sejenis *barcode*. Kode *QR* adalah *barcode* matriks yang dapat dibaca oleh ponsel pintar atau *smartphone* dan ponsel dengan kamera [9]. *QRCode (Quick Respond)* adalah simbol dua dimensi yang merupakan bentuk evolusi dari *barcode* yang merupakan simbol 1 dimensi. Kode *QR* pertama kali diperkenalkan pada tahun 1994 oleh *Denso Wave*, sebuah divisi *Denso Corporation* yang salah satu perusahaan grup utama Toyota di Jepang, dan disetujui sebagai standar internasional ISO (ISO / IEC18004) pada Juni 2000. *Qrcode* dapat menampung data maksimal 7.089 angka [10].

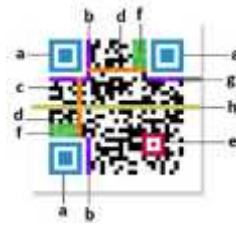
Keunggulan Kode *QR* adalah tidak perlu dipindai dari satu sudut tertentu, karena Kode *QR* dapat dibaca tanpa memperhatikan posisinya. Pemindai kode *QR* mampu menentukan cara yang benar untuk memecahkan kode gambar karena tiga kotak spesifik yang diposisikan di sudut simbol dan blok pelurusan [11].

#### 2.2.1 Karakteristik Qrcode

*Qrcode* memiliki keunggulan lainnya dalam aspek kinerja dan fungsionalitas diantaranya: kecepatan membaca data 360°, fungsi pemulihan data, efisiensi pengkodean karakter kana, fungsi menghubungkan simbol, proses *masking*, kerahasiaan kode dan penandaan secara langsung [10].

### 2.2.2 Struktur *QRcode*

Secara umum bentuk struktur dari *QRcode* seperti gambar 1 berikut [12]:



Gambar 1. Struktur *QRcode*

- a. *Finder Pattern* berfungsi untuk identifikasi posisi *QRcode* dengan menyusun pola terkait pada tiga sudut simbol posisi, ukuran sehingga sudut simbol dapat dideteksi dari segala arah (360 °).
- b. *Format Information* berfungsi untuk informasi tentang *error correction level* dan *mask pattern*.
- c. *Data* berfungsi untuk menyimpan data yang dikodekan.
- d. *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat.
- e. *QRcode* berbentuk modul hitam putih
- f. *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpangan *QRcode* terutama distorsi simbol dengan cara sel terisolasi hitam ditempatkan dalam pola penyalarsan untuk mempermudah mendeteksi koordinat pusat dari pola pelurusan.
- g. *Version Information* adalah versi dari sebuah *QRcode*.
- h. *Quiet Zone* merupakan daerah kosong di bagian terluar *QRcode* yang mempermudah mengenali pengenalan QR oleh sensor CCD.
- i. *QRcode version* adalah versi dari *QRcode* yang digunakan.

### 2.6 *Cloud Storage*

*Cloud storage* merupakan media penyimpanan berbasis *cloud* dengan tingkat keamanan yang terpercaya dan memiliki layanan sinkronisasi yang dapat memungkinkan pengguna mengakses konten digital mereka kapan saja, dari mana saja, dan dengan perangkat apa saja perangkat *smartphone*, *tablet*, atau PC desktop. Fitur pada *cloud* yang digunakan untuk pembuatan aplikasi ini adalah *spreadsheet* yang berfungsi sebagai media penyimpanan dan *app script* yang digunakan untuk media pembuatan *query* [13].

### 2.7 *Spreadsheet*

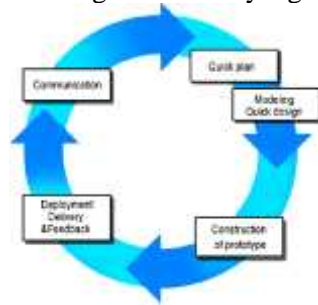
*Spreadsheet* suatu media penyimpanan data yang sangat *up to date* yang mirip dengan media penyimpanan *microsoft excel* tapi media *spreadsheet* ini menggunakan *cloud* sebagai database yang digunakan berulang kali untuk memasukkan kumpulan data yang berbeda. Dalam praktiknya, pengguna biasanya membuat contoh formulir *spreadsheet* yang meniru formulir kertas. *Spreadsheet* berisi data dan rumus tetap, serta data variabel yang berubah dengan setiap kumpulan data. Untuk membuat kumpulan data tambahan, pengguna membuka contoh formulir yang ada, mengubah data variabel, dan menyimpan *spreadsheet* yang dimodifikasi dengan nama *file* baru [14].

### 2.8 *Apps Script*

*App Script* adalah sebuah media yang digunakan untuk menuliskan *code* yang dapat sebagai salah satu diantaranya adalah *connectToInsightly*. Fungsi ini menyetel koneksi ke *Insightly* API. Ini berisi parameter yang diperlukan seperti kunci otentikasi manajer, titik akhir dengan tag [15]. *App script* digunakan dalam penelitian ini untuk menghubungkan kinerja dari *mit app inventor* untuk meng-input data ke *file spreadsheet* sebagai *database* pencatatan dan pelaporan absen.

## 2.1 Kerangka Teori

Dalam Penelitian yang dilakukan, peneliti menggunakan proses model *prototyping*, karena model ini memberi gambaran mengenai sistem yang sudah jadi tapi belum sempurna.



Gambar 2 - Proses Model *Prototyping* [16]

Adapun lima tahap kerangka kerja prototype seperti pada gambar 2, yaitu :

1. *Communication*, merupakan proses awal dengan melakukan pertemuan bersama *stakeholder* untuk menentukan tujuan dari perangkat lunak yang akan dikembangkan dan mengidentifikasi persyaratan apa saja yang diperlukan.
2. *Quick Plan*, merupakan suatu perencanaan secara cepat dalam proses pengembangan sistem sesuai dengan kebutuhan pengumpulan data berupa biaya, waktu dan sumber daya yang akan dibutuhkan dalam fase pengembangan. .
3. *Modeling Quic Design*, merupakan tahap lanjutan dari *Quick Plan* , seorang *developer* melakukan perancangan terhadap sistem sesuai dengan apa yang dibutuhkan oleh *end-user*.
4. *Construction of Prototype*, merupakan suatu tahap untuk memulai pengembangan sistem yang telah sesuai dengan perancangan dan syarat yang diperlukan oleh *user*.
5. *Deployment Delivery and Feedback*, merupakan proses pertimbangan oleh *user* terhadap *software* yang telah dibuat untuk memperjelas kebutuhan pengguna itu sendiri dan selanjutnya akan disesuaikan oleh *developer* hingga kebutuhan *user* terpenuhi.

## 2.2 Sistem yang diusulkan

Berikut pada gambar 3, merupakan kerangka koseptual yang diterapkan dalam pembuatan aplikasi pada penelitian ini :



Gambar 3. Kerangka Konseptual Aplikasi

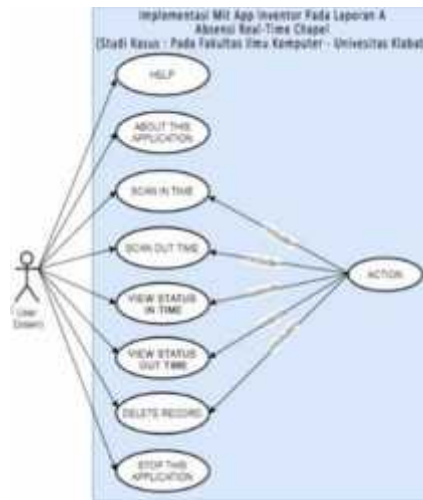
Penjelasan dalam gambar 2.3 sebagai kerangka kerja konseptual aplikasi adalah sebagai berikut:

1. *User* mengaktifkan smartphone dengan sistem operasi android
2. *User* mengaktifkan aplikasi kehadiran real-time *Kuliah Umum*:
  - *User* mengaktifkan aplikasi scan *Qrcode* yang secara otomatis juga mengaktifkan kamera *smartphone*.
  - *User* melakukan pemindaian atau scan *Qrcode* pada *ID card Student*.

- Aplikasi menyimpan data secara otomatis ke spreadsheet (*google drive*) sebagai media penyimpanan *real-time* database sesuai dengan waktu pemindaian.
3. Aplikasi melalui jaringan Internet dapat mengakses data ke ke spreadsheet (*google drive*) sebagai media penyimpanan *real-time* database.

### 2.3 Perancangan Aplikasi

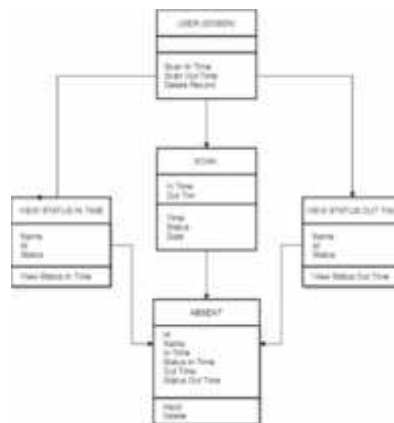
*Use Case* diagram pada gambar 4, digunakan untuk menggambarkan hubungan secara langsung antara pengguna dengan aplikasi. *Use case* diagram terdiri dari *actor* dan *use case*. *Actor* bisa berupa orang atau aplikasi yang terlibat langsung dalam sebuah sistem.



Gambar 4. - *Use case Diagram User*

### 2.4 Class Diagram

*Class Diagram* pada gambar 5 digunakan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan yang terdapat diantara objek-objek tersebut, antara *class* yang satu dengan *class* lainnya pada penulisan Implementasi Mit App Inventor pada Laporan Kehadiran *Real-Time* Kuliah Umum Fakultas Ilmu Komputer – Universitas KLabat.



Gambar 5. *Class Diagram*

Gambar 5 merupakan gambaran hubungan antar class dalam aplikasi. *Class Diagram* juga menunjukkan properti dan operasi dalam sebuah *class* dan batasan-batasan yang terdapat dalam hubungan objek tersebut. Dalam *class diagram* terbagi menjadi 3 bagian, yaitu: nama *class*, *attribute* dan *operation*.

#### 4. HASIL DAN PEMBAHASAN

##### 3.1 MIT App Inventor

Aplikasi ini MIT App Inventor yang menggunakan antarmuka grafis yang memungkinkan pengguna untuk dapat melakukan *drag/drop* objek visual untuk membuat aplikasi yang bisa dijalankan pada perangkat *android* sehingga pengguna tidak perlu lagi menuliskan kode pemrograman untuk menciptakan atau mengembangkan sebuah perangkat lunak pada sistem operasi karena adanya *Future Visual Blocks Programming* yang dapat merubah pengkodean bahasa pemrograman berbasis teks menjadi bahasa visual dalam bentuk *block programming* [17]. Dengan menggunakan bahasa pemrograman yang dibuat menggunakan Google App Script dengan konsep *Cloud Computing* pada akun Google.



Gambar 6. *Interface* Utama MIT App Inventor

Gambar 6 menampilkan antarmuka MIT App Inventor yang berfungsi untuk membuat tampilan antarmuka dari aplikasi Kehadiran *Real-Time Kuliah Umum* ini. Dengan menggunakan MIT App Inventor peneliti memiliki peluang untuk dapat membuat antarmuka sesuai dengan antarmuka yang telah didesain. MIT App Inventor memiliki fitur yang cukup lengkap dan juga lebih mudah dalam hal pengujian aplikasi dengan memanfaatkan emulator *default* yang disediakan dan juga dapat menggunakan emulator yang diinstal pada *smartphone* berbasis Android. Menggunakan Emulator juga cukup mudah hanya dengan memindai QRcode atau dengan cara manual yaitu meng-*input* kode yang ditampilkan secara acak dalam aplikasi emulator. Emulator ini berbasis Android yang dihasilkan dari MIT App Inventor yang dapat diunduh di *Google Playstore*.



Gambar 7. *Blockcodes Interface* MIT App Inventor.

Pada Gambar 7 merupakan konsep pemrograman blok di MIT App Inventor. *Blockcodes* atau *Block Programming* yang disediakan oleh MIT App Inventor terbilang cukup lengkap. Dan dalam hal ini mampu menjawab kebutuhan akan pembuatan aplikasi dalam penelitian ini. Konsep *Block Programming* dilengkapi dengan beberapa fungsi seperti: kontrol, logika, matematika, teks, *list*, warna, variabel, prosedur.



### 3.2 Tampilan Antarmuka Aplikasi Kehadiran *Real-Time Kuliah Umum*

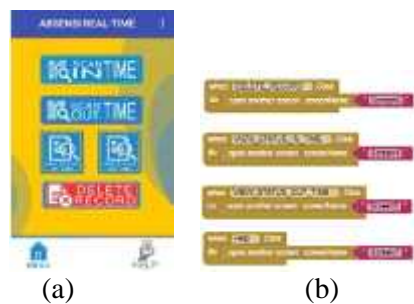
Berikut ini merupakan implementasi perancangan antarmuka aplikasi Kehadiran *Real-Time Kuliah Umum* Fakultas Ilmu Komputer – Universitas Klabat menjelaskan tampilan dari *welcome screen*, menu utama, menu *more*, *help*, *scan in time*, *scan out time*, *view status in time*, *view status out time* dan *delete record*.

#### 3.2.1 Tampilan *Welcome Screen*



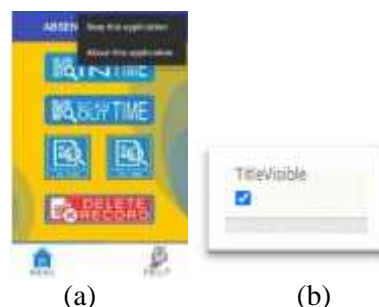
Gambar 8. (a) Tampilan Antarmuka *Welcome Screen* & (b) *Blockcodes Welcome Screen*

#### 3.2.2 Tampilan Menu Utama



Gambar 9. (a) Tampilan Antarmuka Menu Utama & (b) *Blockcodes* untuk masuk ke *screen delete record*, *view status in time*, *view satatus out time*, *help screen*.

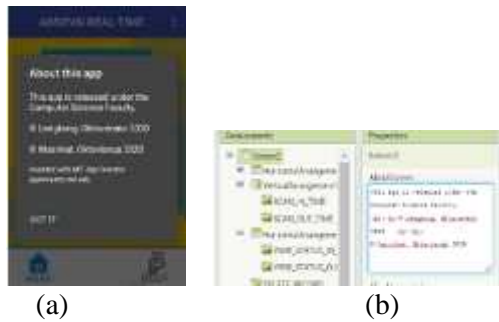
#### 3.2.3 Tampilan Welcome Menu *More*



Gambar 10 (a) Tampilan Antarmuka *More ( Stop & About Application)* & (b) Pengaturan Fungsi Tampilan

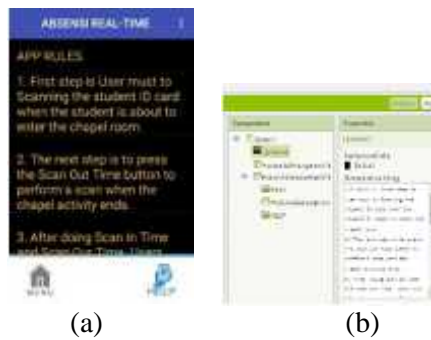


3.2.4 Tampilan *About*



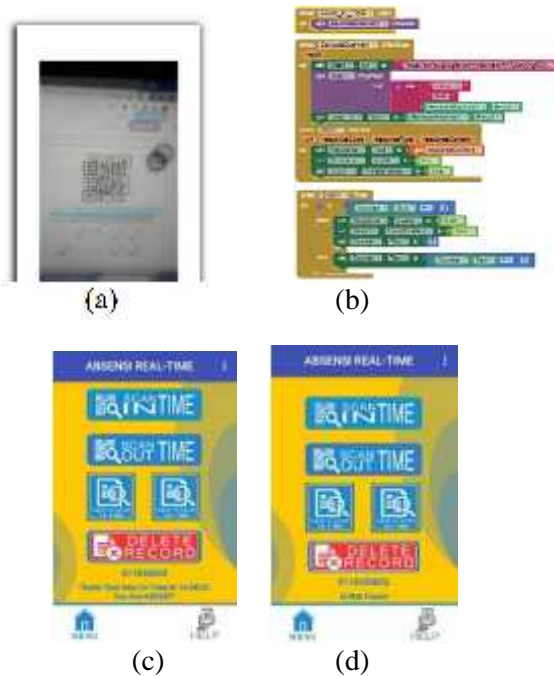
Gambar 11. (a) Tampilan Antarmuka *About* ( *Stop & About Application*) & (b) Pengaturan Tampilan *About*

3.2.5 Tampilan *Welcome Help*



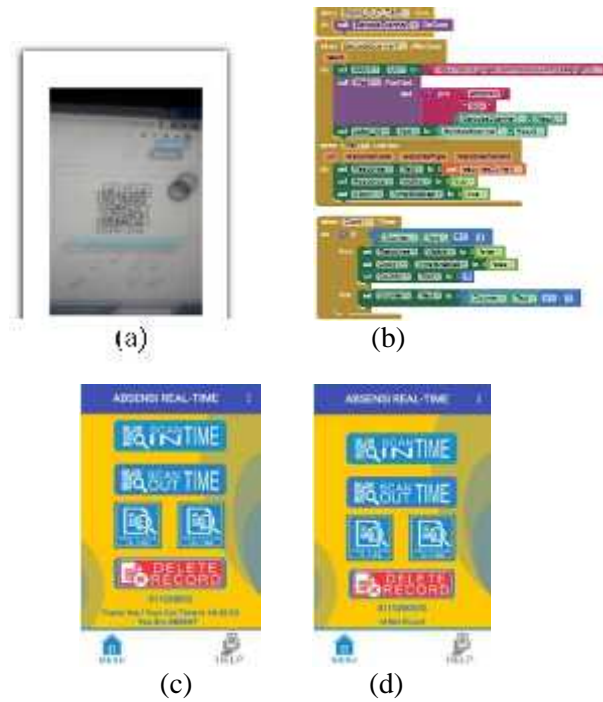
Gambar 12 . (a) Tampilan Antarmuka *Help* ( *Stop & About Application*) & (b) Pengaturan Tampilan *Help*

3.2.6 Tampilan *Scan In Time*



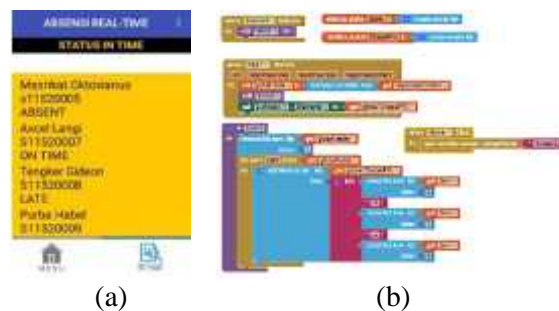
Gambar 13. Tampilan Antarmuka *Scan In Time* – (a) *QRcode Scanning*, (b) *Scan In Time Blockcodes*, (c) *Scanning Succeeded*, (d) *Scanning Failed*

3.2.7 Tampilan *Scan Out Time*



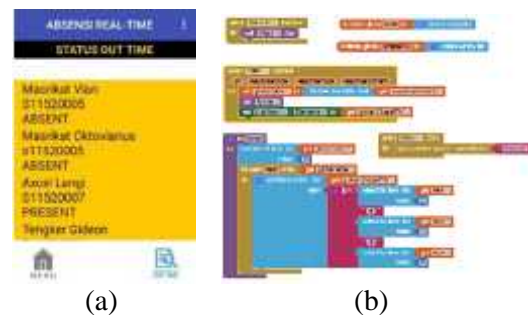
Gambar 14. Tampilan Antarmuka *Scan Out Time* – (a) *QRcode Scanning*, (b) *Scan Out Time Blockcodes*, (c) *Scanning Succeeded*, (d) *Scanning Failed*

### 3.2.8 Tampilan *View Status In Time*



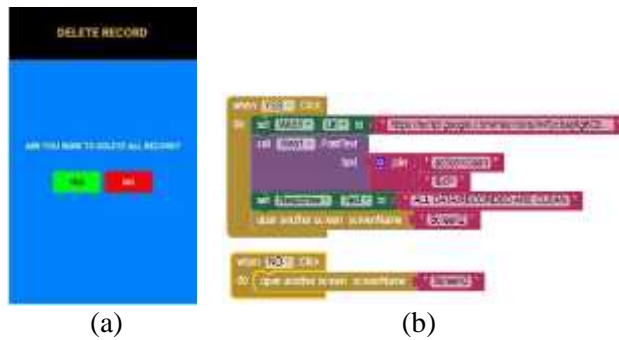
Gambar 15. (a) Tampilan Antarmuka *View Status In Time*, (b) *Blockcodes View Status In Time*

### 3.2.9 Tampilan *View Status Out Time*



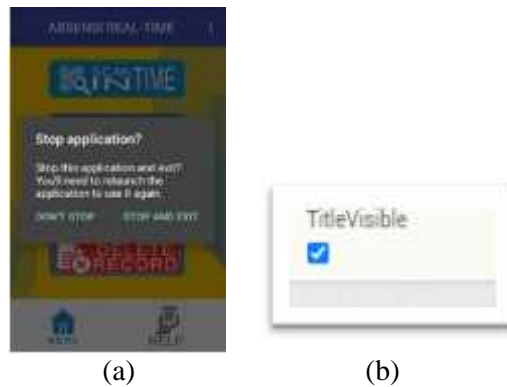
Gambar 3.9 (a) Tampilan Antarmuka *View Status Out Time*, (b) *Blockcodes View Status Out Time*

3.2.10 Tampilan *Delete Record*



Gambar 16. (a) Tampilan Antarmuka *Delete Record*, (b) *Blockcodes Delete Record*

3.2.11 Tampilan *Scan Out Time*



Gambar 17. (a) Tampilan Antarmuka *Stop Application*, (b) Pengaturan Fungsi Tampilan

3.3 Pengujian

3.3.1 Pengujian Fungsi Fitur Aplikasi

Pengujian ini dilakukan untuk mengetahui apakah fungsi dari aplikasi sudah berjalan dengan baik atau tidak. Untuk melakukan pengujian ini peneliti menggunakan metode pengujian *black box*. Metode *black box* merupakan metode pengujian *software* untuk mencari *error* terhadap desain antarmuka dan performa dari aplikasi ini.

Tabel 1. Pengujian Akhir Fitur Aplikasi

<i>Input</i>	<i>Output</i> Yang Diharapkan	Hasil
<i>More</i>	Masuk pada menu <i>More</i>	Berhasil
<i>About</i>	Melihat informasi tentang <i>About</i>	Berhasil
<i>Help</i>	Melihat informasi tentang cara dan alur penggunaan aplikasi	Berhasil
<i>Scan In Time</i>	Menginput data waktu dan status ke <i>database (form absent)</i>	Berhasil

<i>Scan Out Time</i>	Menginput data waktu, status dan tanggal ke <i>database (form absent)</i>	Berhasil
<i>View Status In Time</i>	Menampilkan hasil <i>Scan In Time</i>	Berhasil
<i>View Status In Time</i>	Menampilkan hasil <i>Scan Out Time</i>	Berhasil
<i>Delete Record</i>	Menghapus seluruh data yang di input ke <i>database</i>	Berhasil

Tabel 1 merupakan tabel pengujian fitur yang ada pada aplikasi. Berdasarkan tabel tersebut dapat diketahui bahwa fungsi dari aplikasi sudah berjalan dengan baik dan sesuai dengan output yang diharapkan.

### 3.3.2 Pengujian *Smartphone*

Pengujian ini dilakukan pada 2 jenis perangkat *smartphone* dengan spesifikasi yang berbeda. Pengujian ini dilakukan untuk mengetahui apakah aplikasi dapat terpasang dan berjalan dengan baik di *smartphone* atau tidak.

Tabel 2 Pengujian Akhir pada *Smartphone*

No	Jenis <i>Smartphone</i>	Spesifikasi <i>smartphone</i>	Keterangan
1	Samsung Galaxy J1 Ace	<ul style="list-style-type: none"> <li>• RAM 1 GB</li> <li>• Kamera 5 MP</li> <li>• Android 5.1.1 Lollipop</li> <li>• CPU Quad-core 1.2 GHz Cortex-A7</li> </ul>	Aplikasi terpasang dan berjalan dengan baik
2.	Samsung A10s	<ul style="list-style-type: none"> <li>• RAM 2 GB</li> <li>• Kamera 13 MP</li> <li>• Android 9.0 (Pie)</li> <li>• CPU MSM8917 2GHz</li> </ul>	Aplikasi terpasang dan berjalan dengan baik

Tabel 2 merupakan tabel pengujian pada *smartphone*. Berdasarkan tabel tersebut menjelaskan bahwa aplikasi Kehadiran *Real-Time Kuliah Umum FIK* dapat terpasang dan berjalan dengan baik pada *smartphone* dengan *processor*.

## 4. KESIMPULAN

Berdasarkan Rumusan masalah bagaimana merancang sebuah aplikasi laporan kehadiran *real-time Kuliah Umum* dengan menggunakan pengimplementasian *Mit App Inventor* pada penelitian ini, maka dapat penulis dapat menyimpulkan hal-hal berikut:

1. Dengan demikian membuktikan bahwa, penelitian ini mampu menunjukkan walau hanya menggunakan konsep "*block programming*" yang menjadi konsep yang sebagai penerapan yang ada dalam MIT App Inventor, mampu memberikan alternatif pengembangan aplikasi android untuk fungsi yang cukup kompleks, yang dalam hal ini definisi kompleks mengacu kepada setiap fitur yang mampu dikerjakan oleh aplikasi ini.

2. Aplikasi ini dapat memberikan informasi *real-time* mengenai status absen pada kegiatan Kuliah Umum dari setiap mahasiswa yang terdaftar pada Fakultas Ilmu Komputer – Universitas Klabat.
3. Aplikasi ini dapat dijalankan pada perangkat dengan sistem operasi Android yang didukung dengan koneksi internet yang berfungsi untuk mengirim data atau meng-*input* data ke database (*form absent*).
4. Aplikasi ini dapat mengirim data ke *database* jika *database* juga terkoneksi dengan jaringan internet
5. Aplikasi ini dapat menghapus seluruh data hasil *record* dari database secara *real-time*.

## 5. SARAN

Aplikasi ini masih dapat disempurnakan lagi, yang sekiranya dapat dikembangkan lagi oleh penelitian lanjutan, seperti :

1. Menambahkan fungsi *create sheet* per tanggal pada setiap kegiatan kehadiran *Kuliah Umum*.
2. Menambahkan fungsi edit data yang diinput jika terjadi kesalahan pada pada *form absent* dalam hal ini salah ataupun lupa menginput *student ID* dan nama.
3. Menambahkan fungsi *delete* per data sesuai data yang diseleksi jika terjadi kesalahan.
4. Menambahkan fungsi input manual jika terjadi kerusakan *barcode/QRcode* pada *ID card student*.
5. Menambahkan fungsi *Setting Time Kuliah Umum*.
6. Menambahkan fungsi *login* agar dapat mengetahui *Id user* yang melakukan *record* data.

## DAFTAR PUSTAKA

- [1] S. B. Mir and G. F. Llueca, "Introduction to Programming Using Mobile Phones and MIT App Inventor," *IEEE Rev. Iberoam. Technol. del Aprendiz.*, vol. 15, no. 3, pp. 192–201, Aug. 2020, doi: 10.1109/RITA.2020.3008110.
- [2] Universitas Klabat, "Tentang Kami," 2018. <http://www.unklab.ac.id/id/tentang-kami>.
- [3] R. Firliana, F. Rhohman, and R. W. Purwinanto, "Perancangan Sistem Informasi Absensi Dosen Dengan Validasi Mahasiswa Berbasis Web," vol. 4, no. November, pp. 105–111, 2018.
- [4] O. H. Lengkong, D. H. Fiden, and A. Masrikat, "Sistem Informasi Absensi Real-Time di Universitas Klabat," *CogITO Smart J.*, vol. 2, no. 2, p. 216, 2018, doi: 10.31154/cogito.v2i2.31.216-228.
- [5] I. HUSNA, "Perancangan sistem absensi menggunakan barcode berbasis android," 2017.
- [6] A. Cucus, Y. Aprilinda, I. Sistem, and I. Presensi, "Fakultas Ilmu Komputer," *Sist. ABSENSI DAN PELAPORAN Berbas. Fingerpr. DAN SMS Gatew.*, vol. 7, 2016.
- [7] O. H. Lengkong, "IMPLEMENTATION OF GOOGLE APPS AND MIT APP INVENTOR ON ANDROID BASED REAL TIME ATTENDANCE REPORT APPLICATION," *Abstr. Proc. Int. Sch. Conf.*, vol. 7, no. 1, 2019, doi: 10.35974/isc.v7i1.1601.

- 
- [8] G. Hamdi and Krisnawati, "Membangun Aplikasi Berbasis Android 'Pembelajaran Psikotes' Menggunakan App Inventor," vol. 12, no. 4, pp. 37–41, 2011.
- [9] M. F. Rahman and M. A. Yaqin, "ABSENSI QR CODE BERBASIS e-CONFIRMATION ( BOT TELEGRAM ) DAN e- NOTIFICATION DENGAN TEKNOLOGI GOOGLE ACCESS," vol. 3, pp. 7–12, 2019.
- [10] B. L. Smith, "Qr Codes," *Adult Educ. Museums Art Gall.*, pp. 243–255, 2016, doi: 10.1007/978-94-6300-687-3\_20.
- [11] P. Kieseberg *et al.*, "QR code security," *MoMM2010 - 8th Int. Conf. Adv. Mob. Comput. Multimed.*, pp. 430–435, 2010, doi: 10.1145/1971519.1971593.
- [12] K. M. Cabrera Marino, "APLIKASI TEKNOLOGI QR ( QUICK RESPONSE ) CODE IMPLEMENTASI YANG UNIVERSAL," *Yohana Tri Widayati*, vol. 6, no. 1, pp. 5–9, 2017.
- [13] C. K. Chu, W. T. Zhu, J. Han, J. K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 50–57, 2013, doi: 10.1109/MPRV.2013.72.
- [14] R. W. Corner, J. R. H. Misko, and T. L. Link, "Automatic Spreadsheet Forms," no. 19, 1998.
- [15] K. L. Diaz Arrunategui, "External services integration for internal usage of an organization, using Google App Script," 2015.
- [16] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Edition*. 2010.
- [17] S. S. Prayogo, "RANCANG APLIKASI ANDROID PENGENDALI MOBIL DAN KAMERA MENGGUNAKAN APP INVENTOR," pp. 8–12, 2015.