

PENGEMBANGAN APLIKASI ANDROID MENGGUNAKAN INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) APP INVENTOR 2

Endar Suprih Wihidayat¹, Dwi Maryono²

^{1,2} Pendidikan Teknik Informatika dan Komputer, Universitas Sebelas Maret,
Surakarta, Indonesia
endars@staff.uns.ac.id

Abstrak

Untuk membuat aplikasi berbasis Android, banyak Integrated Development Environment (IDE) yang bisa digunakan. Android Studio dengan bahasa pemrograman Java merupakan IDE dan bahasa pemrograman *native* yang didukung secara resmi oleh Google. Sebagai IDE yang didukung oleh Google, AS menjadi IDE terbaik yang bisa digunakan untuk membangun aplikasi. Namun demikian banyak pemrogram pemula yang merasa kesulitan untuk memulai terutama yang baru memulai konsep pemrograman untuk perangkat *mobile*. Android Studio juga dikenal sebagai IDE yang membutuhkan *requirement* komputer yang handal serta memiliki fitur yang sangat kompleks. Tujuan dari penelitian ini adalah mengembangkan aplikasi yang dibangun dengan IDE yang lebih *user friendly*. lebih menekankan konsep besar dari pemrograman tanpa banyak kesulitan dalam hal menyusun kode program dan juga kompleksitas dari IDE seperti halnya Android Studio. IDE yang digunakan dalam penelitian ini adalah App Inventor 2 (AI2), sebuah IDE yang berbasis *cloud* yang dikelola oleh Massachusetts Institute of Technology (MIT). Tujuan selanjutnya adalah menguji dan mengevaluasi kemampuan AI2 serta aplikasi yang dihasilkan. Dalam penelitian ini telah dikembangkan aplikasi bernama Receptionist yang digunakan untuk menguji kemampuan dari aplikasi yang dihasilkan oleh AI2. Hasil penelitian menunjukkan: (1) Secara umum AI2 menghasilkan aplikasi yang layak serta memiliki usabilitas yang baik (2) AI2 bisa digunakan untuk membangun aplikasi yang dengan fitur kompleks seperti SMS, *query* database dan koneksi database eksternal; (3) produk yang dihasilkan juga memiliki komparabilitas versi Android yang baik; (4) AI2 tidak menyediakan *event handler* untuk fitur yang terkait dengan proses I/O; (5) beberapa fitur seperti pengolahan database yang seharusnya dijalankan di *background services* masih dijalankan di *UI thread* sehingga mengurangi *user experience* saat terjadi *Application not responding* (ANR).

Keywords: App Inventor 2, Aplikasi Android, Cloud Programming, visual programming

Abstract

Many Integrated Development Environment (IDE) can be used to create Android Apps. Android Studio with Java programming language is an IDE and native programming language that is officially supported by Google. As a supported IDE, AS becomes the best IDE that can be used to build apps. Nevertheless, many novice programmers find it difficult to get started especially by starting a new programming concept for mobile devices. Android Studio is also known as an IDE that requires a high specs computer requirement and also has sophisticated features. This research purposed to develop applications with IDEs that are more user-friendly and more emphasize the big concept of programming without much difficulty regarding compiling the program code and also the complexity of IDE as well as Android Studio. The IDE used in this study is App Inventor 2 (AI2), a cloud-based IDE managed by the Massachusetts Institute of Technology (MIT). The next goal is to test the capabilities of the application by conducting Blackbox Testing and User Testing. In this research has been developed an app called Receptionist that is used to test the ability of the codes generated by AI2. The results show that AI2 can be used to build complex applications such as Text Message Service (SMS) broadcasting, performing local database queries and retrieving data on the server. However, some features such as database processing should run in the background still run on the UI thread, thereby reducing the user experience when Application Not Responding (ANR) occurs.

Keywords: App Inventor 2, Android Application, Cloud Programming, visual programming

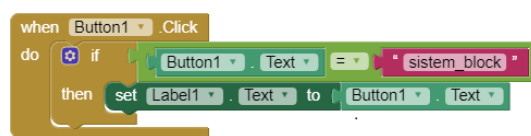
PENDAHULUAN

Aplikasi berbasis *mobile* yang dipasang di *Operating System* (OS) untuk perangkat *mobile* seperti Android, iOS, dan Windows Phone menjadi salah satu jenis aplikasi yang berkembang cepat dewasa ini. Hal ini ditambah dengan tersedianya pasar aplikasi yang mendukung distribusi aplikasi secara mudah. Kondisi diatas mendorong lahirnya banyak *programmer* baru yang mencoba terjun di pemrograman berbasis *mobile*. Mendesain aplikasi yang akan dipasang di perangkat *mobile* jauh lebih sulit dibandingkan dengan aplikasi *desktop*. Hal ini dikarenakan karakteristik dari perangkat *mobile* yang memiliki sejumlah keterbatasan perangkat keras. Secara umum perangkat *mobile* memiliki karakteristik: (1) *Central processing unit* (CPU) dan *Graphical Processor Unit* (GPU) yang terbatas, (2) layar yang kecil, (3) lingkungan kerja yang beragam (*mobile context*), dan (4) koneksi jaringan yang tidak *reliable* (Zhang & Adipat, 2005). Khusus untuk aplikasi berbasis Android, ada banyak ragam IDE tersedia. IDE ini memiliki karakteristik lingkungan kerja yang berbeda (*cloud* versus *desktop*), dan jenis aplikasi yang akan dibuat yang berbeda (aplikasi umum versus game). Google sebagai organisasi yang mengelola Android *men-support* Android Studio (AS) dan bahasa pemrograman Java sebagai IDE dan bahasa pemrograman *native* (asli) dari Android. Dengan *support* langsung dari Google, AS menjadi IDE yang ideal untuk memulai mengembangkan aplikasi. Namun hal ini tidak mudah bagi pengembang baru, dan khususnya yang baru mengenal bahasa pemrograman. Untuk menjembatani gap tersebut, para peneliti di MIT mengembangkan IDE berbasis *cloud* yang mudah digunakan untuk pemula dari sisi pengalaman di pemrograman, maupun dari sisi umur (anak-anak). MIT mengembangkan AI2 sebagai IDE untuk aplikasi Android yang mudah digunakan, berbasis *cloud* dengan kode-kode

berupa *puzzle click & drag*. Dengan metode pemrograman seperti ini pemrograman akan lebih fokus pada level abstraksi dan algoritma program, tidak terpecah pada proses *debugging* dan *listing* program yang rumit.

Lebih lanjut, bagaimanakah kemampuan dari AI2 sebagai IDE, seberapa banyak fitur (atau *class*) yang tersedia yang bisa digunakan. Bagaimana bila ingin membuat aplikasi yang cukup kompleks, apakah AI2 masih mampu memfasilitasi? Pertanyaan tersebut akan dibahas dalam artikel ini. Artikel ini terbagi menjadi beberapa bab. Bab 2 membahas tentang hal-hal berkaitan dengan IDE dari AI2, beberapa fitur yang akan dicoba dalam Receptionist. Bab 3 akan membahas metode yang digunakan dalam penelitian ini, Bab 4 akan membahas hasil dari pengembangan Receptionist dan hasil pengujian, dan Bab 5 berisi kesimpulan dan saran untuk penelitian lebih lanjut.

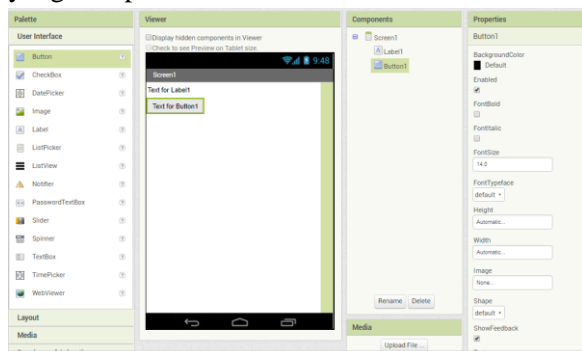
App Inventor 2 (AI2) merupakan IDE generasi kedua dari App Inventor yang dikelola oleh Massachusetts Institute of Technology (MIT). AI2 berbasis *cloud* yang diakses menggunakan internet *browser*. Masuk kategori dalam visual programming, AI2 menggunakan block puzzle yang disusun untuk menjadi rangkaian kode (Gambar 1).



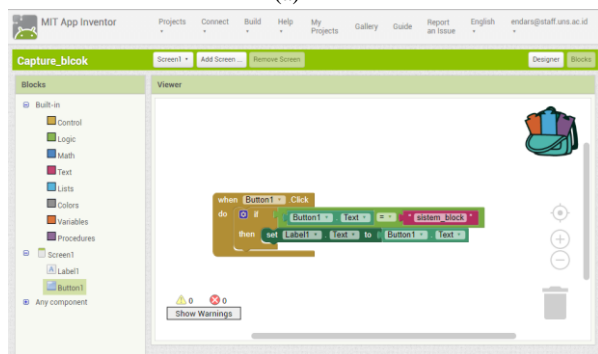
Gambar 1 Block puzzle dalam AI2

AI2 memiliki 3 bagian utama, *Component Designer*, *Block Editor* dan *Android Device* yang digunakan untuk pengujian. Pengujian bisa menggunakan *emulator* maupun perangkat sebenarnya. Untuk perangkat sebenarnya bisa dihubungkan melalui jaringan *wireless* dan menggunakan USB. *Component designer* merupakan *class* dan *method* yang siap digunakan seperti halnya *class* dan *method* dalam bahasa pemrograman Java, hanya saja dalam AI2 dinamakan dengan

komponen (Gambar 2.a). Komponen tersebut adalah: (1) User Interface (2) Layout (3) Media (4) Drawing & Animation (5) Sensor (6) Social Component (7) Storage (8) Connectivity dan (9) Lego MindStorms (Gambar 2.b). *Block Editor* merupakan sekumpulan blok berisi perintah untuk fungsi percabangan, perulangan, variable, array, serta beberapa kelas yang berfungsi seperti *Public Static Class*, jadi kita bisa langsung memakai metode tersebut tanpa perlu instansiasi (membuat objek) terlebih dahulu. Bila dilihat dari komponen yang telah tersedia, AI2 sudah cukup memadai untuk membangun aplikasi yang kompleks.



(a)



(b)

Gambar 2.a.Component Designer dalam AI2, **b.** Block Editor dalam AI2

AI2 selain mudah digunakan juga bisa mengubah persepsi orang terhadap cara membuat *software* (Munoz et al., 2017). Dengan menggunakan AI2 maka pemrogram pemula lebih tertarik untuk belajar membuat aplikasi (Stamatios Papadakis & Orfanakis, 2017). Pada penelitian lain menunjukkan bahwa AI2 bisa membantu siswa sekolah dasar dan sekolah menengah yang baru belajar

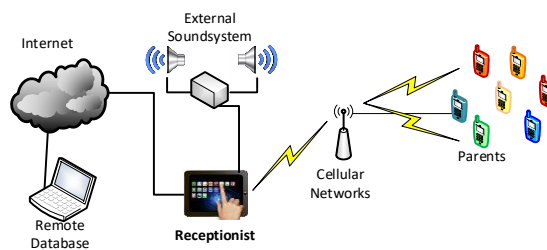
pemrograman dalam mempelajari konsep dasar pemrograman, mengurangi atau bahkan menghilangkan kemungkinan *syntac error* (S. Papadakis, Kalogiannakis, Orfanakis, & Zaranis, 2017).

METODE PENELITIAN

Penelitian dilakukan dalam dua tahap, pertama pengembangan aplikasi Receptionist dan kedua pengujian berupa pengujian *blackbox* dan pengujian usabilitas / tingkat kemudahan pemakaian.

3.1 Pengembangan Aplikasi

Receptionist memiliki fungsi utama sebagai media komunikasi antara siswa atau guru di sekolah dengan orangtua siswa. Alat ini didasari oleh peraturan sekolah yang melarang siswa untuk membawa alat komunikasi sendiri. Hal ini menyebabkan pihak sekolah untuk memfasilitasi siswa dengan media komunikasi. Bentuk komunikasi yang disediakan dengan Receptionist ini adalah SMS, panggilan telepon dan panggilan siswa yang disalurkan ke pengeras suara di sekolah (Gambar 3).



Gambar 3. Arstektur dari Receptionist

Ada banyak sekali komponen yang disediakan oleh AI2, sehingga tidak semua komponen yang tersedia di AI2 digunakan dalam Receptionist. Dalam Tabel 1 ditunjukkan komponen yang akan dimasukkan dalam Receptionist serta kebutuhan fungsional yang sesuai.

Tabel 1. Kebutuhan fungsional Receptionist

Fungsional	Komponen utama	Detail Fungsional	Detail Komponen
SMS Broadcast	-UI -Layout -Social -Storage	SMS to parents, school employee	Texting TinyDB
Phone call /SMS	-UI -Layout -Social -Storage -Sensor	-Phone call to Parents -Phone call to School Employee	-PhoneCall -Texting -TinyDB NFC
Student Call	-UI -Layout -Media -Storage -Sensor	-Students call	-Player -TinyDB -NFC
Data Management	-UI -Layout -Connectivity -Media -Storage Social	-Import Data -Manage Student Data -Manage user -Voice Setting -Web Setting SMS -Setting NFC Setting	-TinyDB Web - ActivityStart r -Player -Texting

Untuk menguji fungsi yang membutuhkan proses lama atau membutuhkan *background process*, Receptionist menggunakan database di server yang dihubungkan melalui internet. Pertama kali dijalankan, database ini harus diunduh untuk disimpan di database lokal (TinyDB). Proses ini akan membutuhkan waktu eksekusi yang cukup lama. Dan proses ini cukup untuk membuktikan apakah AI2 bisa menangani proses eksekusi tanpa mengganggu kenyamanan pengguna. Antarmuka dari Receptionist bisa dilihat di Gambar 4. Pada Gambar 4 terlihat bahwa untuk bisa menampilkan nama kelas dan nama siswa di tombol, dilakukan proses *query* dari database tinyDB yang kemudian ditampilkan hasilnya di Tombol.



Gambar 4. Receptionist user interface

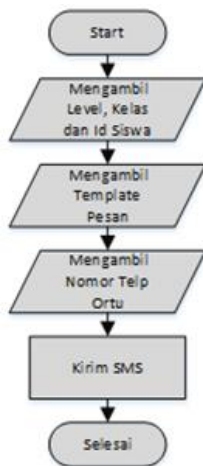
Receptionist seperti dijelaskan dalam Tabel 1 menggunakan fitur NFC sebagai salah satu mode input. NFC digunakan untuk memfasilitasi proses pemanggilan orangtua atau guru ke siswa via pengeras suara di sekolah, juga digunakan untuk jalur cepat menelepon dan SMS siswa ke orangtua. Diperlukan *extended antenna* untuk menaruh *antenna* disamping tablet sehingga pengguna tidak perlu menempelkan kartu NFC di belakang (Gambar 5)



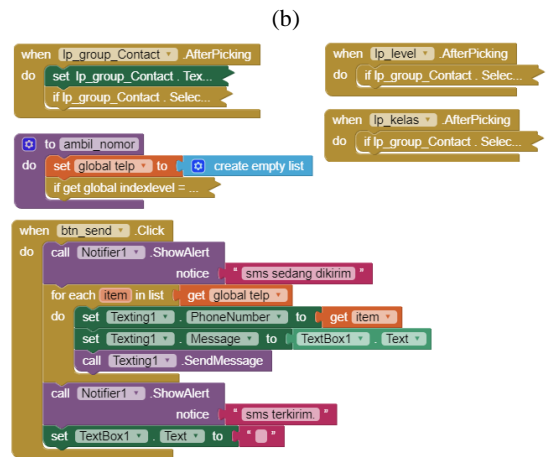
Gambar 5. Receptionist dengan casing dan *extended antenna*

SMS

Fungsi SMS yang dikembangkan dalam Receptionist dibagi menjadi 2, SMS tunggal (dikirim ke 1 nomor) dan SMS broadcast (dikirim ke banyak nomor). Untuk pengiriman SMS tunggal *flowchart* dan potongan kodenya bisa dilihat pada Gambar 6.

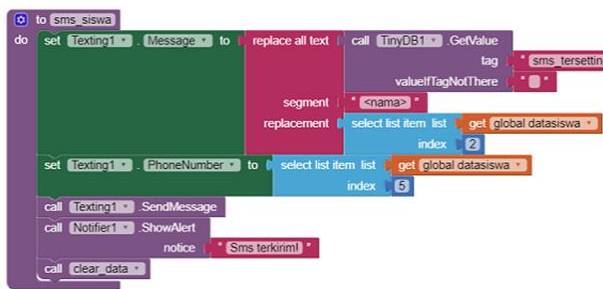


(a)



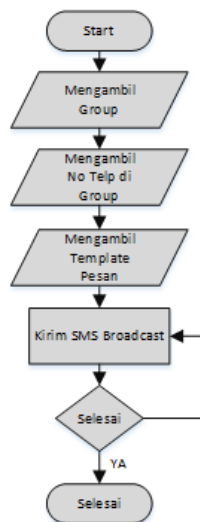
(b)

Gambar 7 (a) Flowchart (b) blok kode untuk SMS Broadcast



(b)

Gambar 6 (a) Flowchart (b) blok kode untuk SMS tunggal

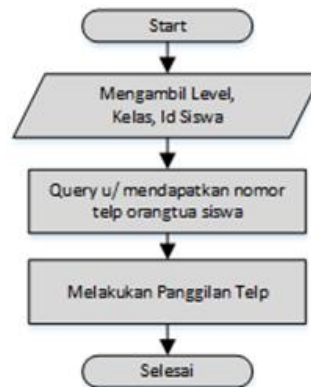


(a)

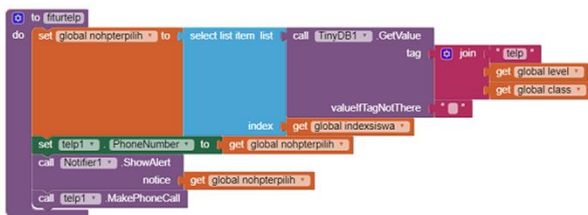
Pada saat proses pengiriman, antara *broadcast* dan pengiriman tunggal, keduanya memiliki proses yang sama, hanya saja, untuk pengiriman SMS *broadcast*, ada proses *looping* berdasarkan jumlah nomor yang dikirim. Nomor yang akan dikirim disimpan dalam bentuk *list* atau *array* (flowchart dan potongan blok bisa dilihat di Gambar 7).

Panggilan Telepon

Untuk melakukan panggilan telepon digunakan komponen Phonecall. Proses sebelumnya sama dengan melakukan SMS tunggal, pilihan Level, Kelas dan Id Siswa selanjutnya digunakan untuk mengambil nomor telp orangtua siswa. Bila sudah diperoleh maka langsung dilakukan fungsi Call. Detail flowchart dan kode bisa dilihat pada Gambar 8.



(a)



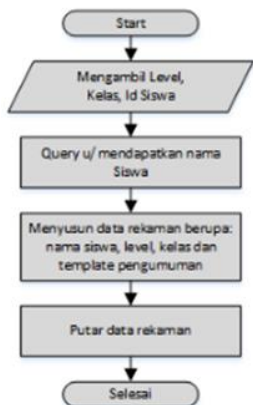
(b)

Gambar 8 (a) Flowchart (b) blok kode untuk melakukan panggilan telepon

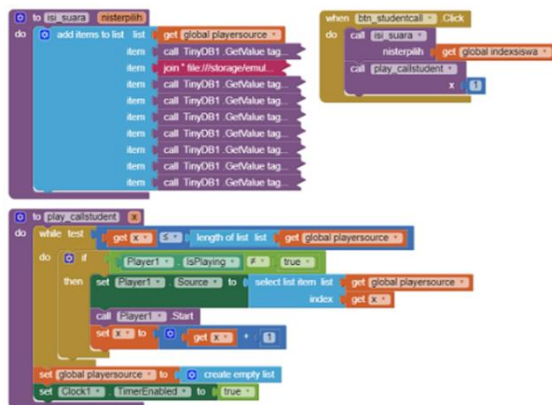
Panggilan Siswa (via external speaker)

Flowchart untuk pemanggilan siswa hampir sama dengan SMS tunggal. Pengguna terlebih dahulu memilih level, kelas, dan id siswa. Dengan 3 parameter tersebut dilakukan proses *query* di database lokal aplikasi (TinyDB) untuk mengambil nama siswa. Dilanjutkan dengan mengisi data untuk diputar di Player berupa rekaman nama level, nama kelas, nama siswa yang sesuai, serta template jenis panggilan yang telah di setting sebelumnya.

Proses akhir adalah menggabungkan kesemua rekaman tersebut untuk di putar menggunakan komponen Player. Detail flowchart dan block yang digunakan bisa dilihat di Gambar 9.



(a)

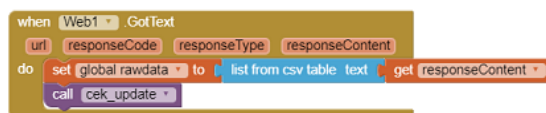


(b)

Gambar 9 (a) Flowchart (b) dan blok kode untuk Panggilan Siswa

Import data

Receptionist menggunakan data yang diambil di server. Hal ini dengan tujuan untuk memudahkan pengelolaan dibandingkan dengan menggunakan data yang langsung di *entry* di perangkat. AI2 menyediakan 2 alternatif media penyimpanan di server/ di cloud, pertama TinyWebDb, dan Fusion Table. Bila kita memiliki sistem informasi yang terlebih dahulu ada, maka kita tidak menggunakan keduanya. Oleh karena itu, Receptionist menggunakan database yang disimpan dalam MySQL. Untuk mengakses, Receptionist menggunakan fasilitas komponen Web, yang merupakan bagian dari Komponen *Connectivity*. Bila menggunakan database sendiri (selain TinyWebDB dan Fusion Table), maka di sistem yang mengelola database harus menyediakan *Application Programming Interface (API)* yang mana menjembatani dalam mengakses ke database. Database yang diimport dari server dalam format Comma Separated Value (CSV) yang kemudian diubah menjadi List (format array dalam AI2) seperti ditunjukkan dalam Gambar 10.



Gambar 10. Block response yang didapat dari server yang kemudian dikonversi ke List

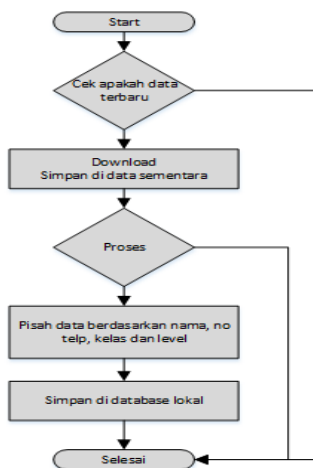
Untuk flowchart lengkap dari proses import data bisa dilihat di Gambar 11.

3.2 Pengujian

Pengujian ditujukan untuk mengetahui kualitas aplikasi. Dalam penelitian ini kualitas yang ingin diukur adalah fungsionalitas dan kemudahan pemakaian. Untuk mengetahui fungsionalitas dilakukan dilakukan pengujian dengan metode *blackbox testing* yang dilakukan oleh pengembang aplikasi. Metode pengujian ini berisi skenario untuk masing-masing fungsi dari aplikasi yang tersedia. Pengujian kedua dilakukan oleh pengguna yang dalam penelitian ini dilakukan oleh 26 siswa SD kelas 6. Partisipan mencoba menjalankan fungsi kemudian mengisi kuisioner untuk mengetahui tingkat kemudahan pemakaian. Pertanyaan diadaptasi dari Software Usability Scale (SUS) yang sudah cukup baik untuk mengukur tingkat kemudahan pemakaian (Sauro & Lewis, 2012). Pada Tabel 2 ditunjukkan daftar pertanyaan yang digunakan yang terdiri dari

HASIL DAN PEMBAHASAN

Pengujian fungsional dilakukan dengan skenario penggunaan setiap fitur pada beberapa jenis *hardware* dan versi Android yang berbeda. Spesifikasi hardware dan versi OS Android bisa dilihat di Tabel 3



Gambar 11. Flowchart proses import data dari server

Tabel 2. Kuisioner untuk mengukur tingkat kemudahan penggunaan aplikasi

No	Pertanyaan
1	Aplikasi berjalan dengan cepat
2	Tidak ditemukan error pada saat dijalankan
3	Tidak ada masalah pada saat pertama kali dibuka
4	Saya sangat yakin apa yang harus dilakukan
5	Tidak dibutuhkan waktu khusus untuk mempelajari
6	Kadang saya tegang saat menjalankan alat
7	Saya tidak membutuhkan teman untuk menjalankan alat ini
8	Saya menikmati saat menggunakan alat ini
9	Aplikasinya menyenangkan
10	Aplikasinya sangat mudah dipahami
11	Saya ingin menggunakan alat ini dalam keseharian
12	Desainnya sangat atraktif
13	Sangat mudah dan cepat untuk pindah halaman
14	Saat dijalankan pertama kali, sangat mudah mengetahui apa yang pertama kali dipencet.

Tabel 3 Detail Spesifikasi hardware dan versi OS Android untuk pengujian

Nama Perangkat	Versi Android	RAM	Prosesor	Jenis	Ukuran Layar (mm)
Google Nexus 7	Marshmallow (6.0)	1 GB	Quad-core 1.2 GHz	Tablet	198.5x120
Sony Xperia M	Jelly Bean (4.1)	1 GB	Dual-core 1.0 GHz	Phone	124x62

Hasil pengujian *blackbox testing* bisa dilihat di Tabel 4

Tabel 4 Hasil pengujian semua fitur Receptionist

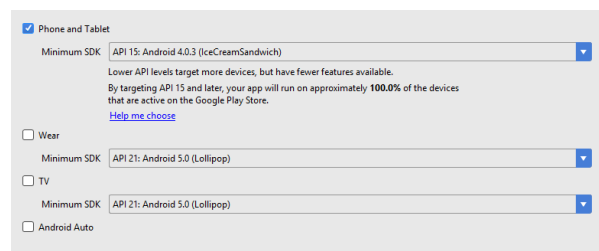
Fungsional	Skenario	Perangkat 1	Perangkat 2
SMS	-SMS tunggal via tombol	Berhasil	Berhasil
	SMS tunggal via NFC	Berhasil	Berhasil
	-SMS Broadcast ke satu kelas	Berhasil	Berhasil
	-SMS Broadcast ke satu Level	Berhasil	Berhasil
	-SMS Broadcast ke satu Sekolah	Berhasil	Berhasil
	Phone call	-Phone call via tombol	Berhasil
	-Phone call via NFC	Berhasil	Berhasil
Student Call	-Panggilan via tombol	Berhasil	Berhasil
	-Panggilan via NFC	Berhasil	Berhasil
Data Management	-Setting Rekaman	Berhasil	Berhasil
	-Setting Web	Berhasil	Berhasil
	-Import Data dari Server	Berhasil	Berhasil
	-Update User	Berhasil	Berhasil
	-Update template SMS		
	-Update template Rekaman		

4.1 Versi Android

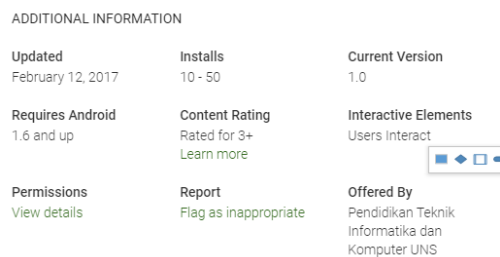
AI2 tidak secara eksplisit menjelaskan target versi Android dari aplikasi yang dikembangkan. Hal ini berbeda dengan IDE Android Studio yang menentukan target versi Android di awal pengembangan aplikasi (Gambar 12)

Semakin luas cakupan versi Android dimana aplikasi bisa dijalankan maka semakin baik karena menjangkau semakin banyak pengguna. Namun disisi lain, aplikasi tersebut semakin sulit untuk dibuat karena harus menambahkan kode agar *compatible* dengan

beberapa versi, terutama untuk versi lama. Hasil APK yang dibuat dengan AI2 setelah di upload di Google Play terlihat bahwa versi Android paling minimal adalah versi 1.6 (Android Donut). Hal ini menjadi salah satu kelebihan, karena AI2 menghasilkan aplikasi dengan range versi OS yang luas, walaupun perangkat keras yang masih menggunakan Android versi 1.6 boleh dibilang sudah tidak ada.

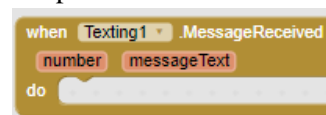


Gambar 12. Setting minimum SDK pada Android Studio



Gambar 13. Minimum versi Android dari AI2 setelah diupload di Google Play

Bila tablet tidak memiliki bawaan aplikasi SMS maka harus dipasang terlebih dahulu aplikasi SMS pihak ketiga dari Google Play. Untuk *event handler*, AI2 hanya menyediakan fungsi untuk menerima SMS yang masuk (Gambar 14). Tidak ada event atau fitur untuk mengecek apakah SMS berhasil atau gagal dikirim, misal dikarenakan jaringan yang jelek. Bahkan dalam mode terbang, perintah SMS di AI2 akan tetap di eksekusi.



Gambar 14 Event handler SMS masuk dengan variable Number dan MessageText

Tabel 5. Class dari yang bisa digunakan untuk manajemen SMS.

Nested Class	Keterangan
Telephony.Sms.Conversations	Contains all sent text-based SMS messages in the SMS app.
Telephony.Sms.Draft	Contains all sent text-based SMS messages in the SMS app.
Telephony.Sms.Inbox	Contains all text-based SMS messages in the SMS app inbox.
Telephony.Sms.Intents	Contains constants for SMS related Intents that are broadcast.
Telephony.Sms.Outbox	Contains all pending outgoing text-based SMS messages.
Telephony.Sms.Sent	Contains all sent text-based SMS messages in the SMS app.

4.2 Fitur SMS

Menggunakan Android Studio yang berbasis Java, SMS bisa dibuat dengan 2 metode, menggunakan SmsManager API dan dengan Built-in SMS *application* atau melimpahkan aktivitas ke aplikasi SMS bawaan yang sudah tersedia. Bila dibandingkan, maka banyak fitur di AI2 yang tidak tersedia yang sebenarnya diperlukan seperti fungsi pengelolaan SMS seperti Telephony.Sms.Conversations, .Draft, .Inbox, .Outbox, .dan .Sent (Tabel 5). AI2 hanya menyediakan fitur SMS dengan memanggil built-in SMS *application* yang menjadi bawaan *handphone/tablet*.

4.3 Panggilan Telepon

Sama halnya dengan fitur SMS, AI2 tidak memiliki fitur untuk melakukan panggilan telepon untuk panggilan telepon dari aplikasi yang dibuat secara langsung, melainkan harus melalui *built-in application* bawaan dari Android. Untuk memanggil *built-in application* AI2 menggunakan komponen PhoneCall. Karena menggunakan *built-in application*, tidak semua fitur atau event bisa ditangkap oleh Receptionist.

Event seperti STATE_CONNECTING, STATE_DIALING, STATE_DISCONNECTING yang disediakan oleh bahasa *native* (Java) tidak tersedia. AI2 hanya memperoleh status terputus yang otomatis kembali ke aplikasi Receptionist.

4.4 Import data dan pengolahan lokal

Database lokal yang disediakan oleh AI2 salah satunya adalah TinyDb. TinyDB tidak mengenal konsep tabel namun hanya “tag” yang seperti halnya nama kolom dalam tabel. Setiap tag hanya menyimpan satu record atau satu baris. Hal ini tentu menyulitkan bila data yang tersimpan terdiri dari banyak kolom dan banyak baris. Penelitian ini menggunakan data dari Sekolah SDII Al Abidin yang memiliki 1185 siswa. Terdiri dari 6 level dan 43 kelas. Masing-masing level terdiri dari 6 sampai 8 kelas. Tabel terdiri dari 10 kolom yaitu:

1. NIS (nomor induk siswa)
2. NISN (nomor induk siswa nasional)
3. Nama Siswa
4. Nama Panggilan Siswa
5. Nama Ayah
6. No HP Ayah
7. Nama Ibu
8. No HP Ibu
9. Kelas
10. Nama Suara (nama file suara untuk masing-masing anak)

Untuk bisa menyimpan tabel dalam TinyDB maka harus disimpan dalam bentuk List. List tidak lain adalah *array* dalam format AI2. Yang membedakan, di AI2 index pertama dari List dimulai dari 1, tidak dari 0 seperti halnya umumnya peng-indeks-an dalam array. Karena masing-masing siswa terdiri dari beberapa kolom maka tidak cukup satu lapis List, melainkan minimal dua lapis. Lapis List pertama berisi kumpulan data profil masing-masing siswa, lapisan kedua berisi kumpulan data siswa. Bila ditampilkan, bentuk List yang disimpan dalam TinyDB bisa dilihat sebagai berikut:

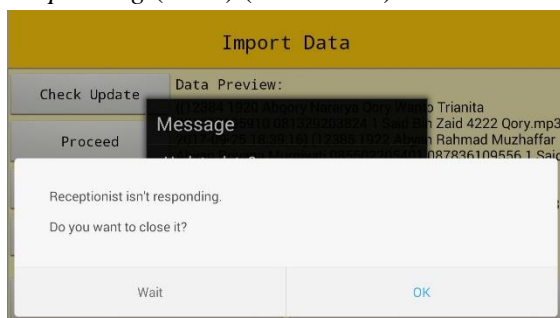
((NIS1 NISN1 Nama_Siswa1 Nama_Panggilan1 Nama_Ayah1 No_HP_Ayah1 Nama_Ibu No_HP_Ibu1 Kelas Nama_Suara) (NIS2 NISN2 Nama_Siswa2 Nama_Panggilan Nama_Ayah No_HP_Ayah Nama_Ibu No_HP_Ibu Kelas Nama_Suara) dst...)

Kurung awal pertama merupakan list untuk satu sekolah, kurung awal kedua list untuk masing-masing siswa. Data dalam list dipisah menggunakan spasi. Pada saat menyimpan dalam List, bila tipe data String maka harus menggunakan tanda petik double. Data yang diambil dari server masih dalam format Comma separated value (CSV) seperti ditunjukkan dalam Gambar 15.

1,1671,Abdul Halim,Halim,Agus Purwanto,Aminah Zahriyati,081548655744,081327687879,1,AMRU BIN ASH,1564,Halim.mp3,2017-05-20 11:53:59 2,1673,Adam

Gambar 15. Format CSV output dari request data server.

Data dalam format CSV tersebut perlu dikonversi ke dalam format List. Proses ini memerlukan waktu untuk eksekusi. Dan di dalam AI2 tidak disediakan fasilitas untuk membuat proses eksekusi dijalankan di *background* sebagaimana di Java *native* sehingga pada saat proses eksekusi bila ada interaksi yang dilakukan oleh user akan memunculkan dialog *Application Not Responding* (ANR) (Gambar 16).



Gambar 16. ANR pada saat proses pengkonversian data

Dijelaskan dalam *Android Reference* bahwa ANR akan muncul dalam dua kondisi, pertama bila aplikasi tidak merespon *event input* lebih dari 5 detik, kedua bila sebuah *BroadcastReceiver* tidak dieksekusi dalam waktu 10 detik. Dalam AI2 untuk menghindari

hal tersebut terjadi disarankan untuk memanfaatkan *timer* melalui fitur Clock dengan memecah proses yang lama menjadi beberapa proses yang lebih kecil. Dengan timer ini, aplikasi akan memecah atau membuat beberapa tahapan proses dari sebuah proses besar.

4.5 Pengujian pengguna

Pengujian alat juga telah dilakukan dengan melibatkan siswa Sekolah Dasar di SD Islam Internasional (SDII) Al Abidin dengan jumlah peserta sebanyak 26 anak. Siswa diarahkan untuk menjalankan alat, kemudian di akhir pengujian siswa mengisi kuisioner dengan pertanyaan seperti ditunjukkan dalam Tabel 2. Jawaban kuisioner diukur menggunakan skala Likert yang memiliki range antara 1 sampai 5, dengan 1 memiliki arti Sangat Tidak Setuju, 2 Tidak Setuju, 3 Tidak Tahu, 4 Setuju dan 5 berarti Sangat Setuju. Hasil kuisioner bisa dilihat di Tabel 5

Tabel 5 Pertanyaan kuisioner dan hasilnya

Pertanyaan	NFC	Button
Aplikasi berjalan dengan cepat	4.8	4.577
Tidak ditemukan error pada saat dijalankan	4.2	4.385
Tidak ada masalah pada saat pertama kali dibuka	4.4	4.462
Saya sangat yakin apa yang harus dilakukan	4.3	4.346
Tidak dibutuhkan waktu khusus untuk mempelajari	4.2	4.115
Kadang saya tegang saat menjalankan alat	3.1	3.423
Saya tidak membutuhkan teman untuk menjalankan alat ini	4.3	4.231
Saya menikmati saat menggunakan alat ini	4.074	3.962
Aplikasinya menyenangkan	4.037	4.077
Aplikasinya sangat mudah dipahami	4.519	4.346
Saya ingin menggunakan alat ini dalam keseharian	3.296	3.462
Desainnya sangat atraktif	3.444	3.462
Sangat mudah dan cepat untuk pindah halaman	4.111	4.154
Saat dijalankan pertama kali, sangat mudah mengetahui apa yang pertama kali dipencet.	4.148	4.5

Dari hasil pengukuran tersebut bisa diketahui rata-rata keseluruhan adalah 4.11. Bila dianalisa lagi usabilitas dengan menggunakan rumus

$$\text{Persentase} = \text{Frekuensi} \frac{\text{Frekuensi yang diperoleh}}{\text{Frekuensi yang diharapkan}} \times 100\%$$

Maka diperoleh angka 82.29 %. Dengan pengkategorian (Sugiyono, 2009) seperti ditunjukkan dalam Tabel 6, maka bisa disimpulkan aplikasi Receptionist sangat mudah digunakan.

Tabel 6 Presentasi Kelayakan

Skor	Kategori Usabilitas
81%-100%	Sangat Mudah
61%-80%	Mudah
41%-60%	Cukup Mudah
21%-40%	Tidak Mudah
<20%	Sangat Tidak Mudah

KESIMPULAN DAN SARAN

Kesimpulan yang bisa diambil dari penelitian ini adalah:

1. Secara umum hasil aplikasi dari AI2 sudah cukup baik dan memiliki tingkat usabilitas dengan status sangat mudah digunakan, hal ini menunjukkan AI2 bisa menghasilkan produk yang layak.
2. Hasil pengembangan Receptionist menunjukkan AI2 bisa digunakan untuk membangun aplikasi dengan fitur yang cukup kompleks seperti untuk SMS *broadcast*, memutar rekaman suara yang dengan jumlah rekaman yang cukup banyak, memproses database dengan jumlah lebih dari 1000 baris.
3. Range versi Android yang dihasilkan oleh IDE AI2 sangat luas, dimulai dari versi awal Android, 1.6 atau Android Donut. Hal ini merupakan hal positif namun sekaligus mengurangi *user experience* misal tampilan yang sederhana tanpa material *design* yang bagus.
4. Untuk fungsi yang memerlukan jaringan baik WIFI ataupun seluler, tidak ada fitur dari AI2 untuk mengecek apakah *handphone* sedang dalam kondisi *online*

atau terhubung. Hal ini diperlukan untuk menghindari *error* bila ingin melakukan eksekusi seperti SMS, telepon ataupun *import* data dari internet. Jadi karena AI2 tidak menyediakan fitur tersebut sebagai akibatnya proses SMS, panggilan telepon dan *import* data dimungkinkan akan menghasilkan *error* tanpa bisa dicegah sebelum melakukan eksekusi.

5. AI2 tidak memiliki kemampuan *background service* sehingga fitur yang membutuhkan proses lebih dari 1 detik akan mengurangi *user experience*. Hal ini bisa dikurangi dengan memecah proses besar menjadi beberapa proses kecil menggunakan komponen sensor *clock*. *Clock* ini menghasilkan *event trigger* yang bisa digunakan untuk menggabungkan proses yang telah di pecah.

DAFTAR PUSTAKA

- Munoz, R., Barcelos, T., Villarroel, R., Becerra, C., Noel, R., & Ponce, F. (2017). Using app inventor to change perceptions about software engineering. *Proceedings - International Conference of the Chilean Computer Science Society, SCCC*.
<https://doi.org/10.1109/SCCC.2016.7836023>
- Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, 12(4), 58–77.
<https://doi.org/10.4018/IJWLTT.2017100106>
- Papadakis, S., & Orfanakis, V. (2017). The Combined Use of Lego Mindstorms NXT and App Inventor for Teaching Novice Programmers. *International Conference on Educational Robotics, EDUROBOTICS 2016*, 560.
<https://doi.org/10.1007/978-3-319-55553-9>

- Sauro, J., & Lewis, J. (2012). *Quantifying the User Experience Practical Statistics for User Research*. Elsevier Inc.
- Sugiyono. (2009). *Metode Penelitian Kuantitatif, Kualitatif dan RnD* (8th ed.). Bandung: Penerbit Alfabeta.
- Zhang, D., & Adipat, B. (2005). Challenges, Usability, Methodologies, and Issues in the Usability Testing of Mobile Applications. *International Journal of Human-Computer ...*, 18(3), 293–308. <https://doi.org/10.1207/s15327590ijhc1803>