

PERANCANGAN APLIKASI TRY OUT UJIAN NASIONAL DENGAN ALGORITMA FISHER-YATES PADA BIMBINGAN BELAJAR ANASTASIA BERBASIS ANDROID

Rudi Santoso¹, Genrawan Hoendarto², Lina³

^{1,2,3}Teknik Informatika, STMIK Widya Dharma, Pontianak

e-mail: ¹rudi_santoso88@yahoo.com, ²genrawan@yahoo.com, ³linalo_77@yahoo.com

Abstract

Anastasia Learning Guidance is currently conducting try out national exam for class XII students by using the manual way that is through the paper sheet of questions and answers. This certainly has many disadvantages such as spending money on paper purchases, takes a long time for the correction of try out results, and cheating by students when try out activities take place. Therefore, it needs a try out application that is online and easy to use by students. In designing this try out application, the author studied the literature related to Fisher-Yates algorithm and programming related to Android. Data collection techniques used were interviews, observation, and literature study. The technique of analysis and system design used is object oriented technique. The modeling tool used is Unified Modeling Language (UML). The design of this try out application using Android Studio to design software, Firebase Realtime Database for database management, and Firebase Cloud Storage as file storage. This study resulted in a try out national exam application that aims to overcome the existing weaknesses in try out activities manually on Anastasia Learning Guidance. Based on the research conducted, it is concluded that the try out national exam application can improve and optimize the implementation of try out activity on Anastasia Learning Guidance. The use of the Fisher-Yates algorithm is capable of randomizing questions so that each student gets different questions from each other.

Keywords: Try Out, National Exam, Fisher-Yates Algorithm, Android

Abstrak

Bimbingan Belajar Anastasia saat ini melaksanakan kegiatan try out ujian nasional bagi siswa kelas XII dengan menggunakan cara manual yaitu melalui lembar kertas soal dan jawaban. Hal ini tentu mempunyai banyak kelemahan seperti menghabiskan biaya untuk pembelian kertas, memakan waktu yang lama untuk pengoreksian hasil try out, dan kecurangan yang dilakukan siswa seperti menyontek saat kegiatan try out berlangsung. Oleh sebab itu, diperlukan sebuah aplikasi try out yang bersifat online serta mudah digunakan oleh siswa. Dalam perancangan aplikasi try out ini, penulis mempelajari literatur-literatur yang berhubungan dengan algoritma Fisher-Yates dan pemrograman yang berhubungan dengan Android. Teknik pengumpulan data yang digunakan adalah wawancara, observasi, serta studi kepustakaan. Teknik analisis dan perancangan sistem yang digunakan yaitu teknik berorientasi objek. Alat pemodelan yang digunakan adalah Unified Modeling Language (UML). Perancangan aplikasi try out ini menggunakan Android Studio untuk merancang perangkat lunak, Firebase Realtime Database untuk manajemen basis data, serta Firebase Cloud Storage sebagai tempat penyimpanan file. Penelitian ini menghasilkan sebuah aplikasi try out ujian nasional yang bertujuan untuk mengatasi kelemahan yang ada dalam kegiatan try out secara manual pada Bimbingan Belajar Anastasia. Berdasarkan penelitian yang dilakukan, didapat kesimpulan bahwa aplikasi try out ujian nasional dapat meningkatkan dan mengoptimalkan pelaksanaan kegiatan try out pada Bimbingan Belajar Anastasia. Penggunaan algoritma Fisher-Yates mampu melakukan pengacakan soal sehingga setiap siswa memperoleh soal-soal yang berbeda satu sama lain.

Kata Kunci: Try Out, Ujian Nasional, Algoritma Fisher-Yates, Android

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah membawa berbagai perubahan dalam aktivitas manusia. Berbagai aktivitas mengalami perubahan dari cara konvensional yang secara praktiknya manual menjadi cara modern yang umumnya berbentuk digital. Masyarakat lebih memilih untuk menggunakan cara modern karena menawarkan kecepatan, ketepatan, dan kemudahan dalam melakukan kegiatan sehari-hari.

Salah satu teknologi informasi yang berkembang pesat saat ini adalah teknologi *mobile* dengan sistem operasi Android. Hampir semua kalangan masyarakat memiliki *smartphone* untuk berkomunikasi, mendapatkan hiburan, serta memperoleh informasi yang cepat dengan dukungan koneksi internet. Selain itu, penggunaan *smartphone* berbasis Android juga banyak dimanfaatkan dalam berbagai lembaga, salah satu contohnya adalah lembaga pendidikan.

Bimbingan Belajar Anastasia merupakan lembaga pendidikan yang turut membantu meningkatkan prestasi akademik siswa di sekolah serta mempersiapkan siswa untuk menghadapi ujian nasional. Salah satu cara yang dilakukan adalah melaksanakan kegiatan *try out* sehingga siswa dapat mengetahui tingkat kemampuan dan kesiapannya dalam menghadapi ujian nasional. Selain itu, dengan adanya kegiatan *try out*, staff pengajar juga dapat mengetahui pemahaman materi siswanya serta dapat melakukan evaluasi proses mengajar.

Pada saat ini, Bimbingan Belajar Anastasia masih menggunakan cara konvensional dalam melakukan kegiatan *try out* bagi siswanya yaitu melalui lembar kertas soal beserta lembar jawaban yang diberikan ke setiap siswa. Hal tersebut tentu masih banyak kekurangannya, yaitu pemborosan kertas, memakan banyak waktu dalam proses penilaian, dan kecurangan yang dilakukan siswa seperti menyontek jawaban dari siswa yang lain. Oleh karena itu, dibutuhkan suatu aplikasi pada *smartphone* yang sudah terintegrasi secara *online* sebagai solusi dalam mengoptimalkan kegiatan *try out* tersebut, dengan mengingat bahwa hampir setiap siswa telah menggunakan *smartphone*. Untuk meminimalisir tingkat kecurangan yang dilakukan oleh siswa, salah satunya adalah dengan memiliki fitur pengacakan soal pada aplikasi sehingga soal yang didapatkan oleh setiap siswa berbeda satu sama lain. Oleh sebab itu, fitur pengacakan soal dapat menggunakan algoritma Fisher-Yates. Algoritma Fisher-Yates merupakan sebuah algoritma untuk menghasilkan permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak himpunan tersebut.

Berdasarkan uraian di atas, maka penulis ingin merancang aplikasi *try out* ujian nasional dengan algoritma Fisher-Yates pada Bimbingan Belajar Anastasia berbasis *mobile* yang dapat dijalankan di *smartphone* dengan sistem operasi Android. Pemilihan aplikasi *mobile* berbasis Android ini dikarenakan hampir semua kalangan pelajar terutama siswa di Bimbingan Belajar Anastasia telah menggunakan *smartphone* dalam kesehariannya. Aplikasi ini diharapkan dapat digunakan sebagai solusi yang tepat untuk mengoptimalkan kegiatan *try out* pada siswa di Bimbingan Belajar Anastasia. Selain itu, aplikasi juga memungkinkan setiap siswa untuk berdiskusi dengan staff pengajar maupun antar sesama siswa lain terkait jawaban atau pembahasan *try out*.

2. METODE PENELITIAN

2.1 Rancangan Penelitian, Teknik Pengumpulan Data, Teknik Analisis Sistem, dan Aplikasi Perancangan Sistem.

2.1.1 Rancangan Penelitian

Dalam penyusunan skripsi ini, penulis menggunakan desain penelitian deskriptif karena menggambarkan kejadian secara nyata yang sesuai dengan keadaan di lapangan pada saat penelitian dilakukan.

2.1.2 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan penulis dalam penelitian ini adalah sebagai berikut:

2.1.2.1 Wawancara

Metode ini dilakukan dengan mengajukan pertanyaan-pertanyaan kepada pemilik Bimbingan Belajar Anastasia yang berhubungan dengan topik yang dibahas.

2.1.2.2 Observasi

Metode ini dilakukan dengan melakukan pengamatan secara langsung terhadap pelaksanaan kegiatan *try out* pada Bimbingan Belajar Anastasia.

2.1.2.3 Studi Kepustakaan

Metode ini dilakukan dengan cara mencari dan mengumpulkan data yang dapat mendukung penulis dalam menyelesaikan skripsi ini. Data dapat berupa teori-teori, konsep-konsep yang berasal dari berbagai media kepustakaan, seperti buku, jurnal, *e-book*, dan lain-lain.

2.1.3 Teknik Analisis Sistem

Teknik analisis sistem yang digunakan adalah teknik berorientasi objek. Alat pemodelan sistem yang digunakan adalah *Unified Modeling Language (UML)*, yang berperan untuk membantu menggambarkan prosedur dan aliran data yang terdapat pada perancangan aplikasi *try out* ujian nasional berbasis Android.

2.1.4 Aplikasi Perancangan Sistem

Aplikasi perancangan sistem yang digunakan dalam penelitian ini adalah menggunakan pemrograman *mobile* dengan Android Studio versi 3.0.1 untuk merancang perangkat lunak, *Firebase Realtime Database* untuk manajemen basis data, serta *Firebase Cloud Storage* sebagai tempat penyimpanan *file*.

2.2 Landasan Teori

2.2.1 Sistem Operasi

Sistem operasi adalah perangkat lunak yang berfungsi mengaktifkan seluruh perangkat yang terpasang pada komputer sehingga masing-masingnya dapat saling berkomunikasi. [1] Sistem operasi adalah sebuah perangkat lunak sistem komputer yang membantu perangkat keras dalam menjalankan fungsi-fungsi manajemen proses. [2]

2.2.2 Program Aplikasi

Program aplikasi merupakan perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows, permainan (*game*), dan sebagainya. [3] Perangkat lunak aplikasi (*application software*) terdiri dari beberapa program yang didesain untuk membuat penggunaanya lebih produktif dan/atau untuk membantu mereka dalam melakukan tugas. [4]

2.2.3 Perancangan Sistem

Tahapan perancangan subsistem-subsistem memungkinkan analis/perancang sistem melakukan upaya untuk mengorganisasi bagian-bagian model perancangan menjadi bagian-bagian yang lebih mudah dipahami dan lebih mudah dikelola. [5] Rancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru. Jika sistem itu berbasis komputer, rancangan dapat menyertakan spesifikasi jenis peralatan yang akan digunakan. [6]

2.2.4 Perancangan Antarmuka Pengguna

Teknik perancangan antar muka melibatkan berbagai komponen di dalamnya, seperti pemahaman terhadap *human factor* atau faktor manusia di dalam berinteraksi dengan aplikasi yang akan dibuat. [7] Perancangan antarmuka pemakai berkaitan dengan studi terhadap manusia yaitu pemakai dari perangkat lunak yang dibangun. [8]

2.2.5 Try Out

Try Out adalah salah satu bentuk evaluasi yang dilakukan sekolah dan pihak terkait untuk mempersiapkan siswa menghadapi UN. [9]

2.2.6 Ujian Nasional

Ujian Nasional (UN) sebagai bentuk penilaian hasil belajar oleh pemerintah yang menjadi tolak ukur keberhasilan siswa dalam mengikuti proses pembelajaran di sekolah. [9] Ujian Nasional adalah sistem evaluasi standar pendidikan dasar dan menengah secara nasional dan persamaan mutu tingkat pendidikan antar daerah yang dilakukan dengan menetapkan standarisasi nasional pendidikan yang bertujuan sebagai pemetaan masalah pendidikan dalam rangka menyusun kebijakan pendidikan nasional. [10]

2.2.7 Algoritma Fisher-Yates

Algoritma Fisher-Yates merupakan metode pengacakan yang lebih baik, dengan waktu eksekusi yang cepat serta tidak memerlukan waktu yang lama untuk melakukan suatu pengacakan. Algoritma Fisher-Yates terdiri dari dua metode, yaitu metode *original* dan metode *modern*. Metode *modern* dipilih karena metode ini memang khusus digunakan untuk pengacakan dengan sistem komputerisasi karena hasil pengacakan bisa lebih variatif. [11] Algoritma Fisher-Yates yang diambil dari nama Ronald Fisher dan Frank Yates adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak himpunan tersebut. Metode dasar yang digunakan untuk menghasilkan suatu permutasi acak untuk angka 1 sampai N adalah sebagai berikut: (a) tuliskan angka 1 sampai N, (b) pilih sebuah angka acak K diantara 1 sampai dengan jumlah angka yang belum dicoret, (c) dihitung dari bawah, coret angka K yang belum dicoret dan tuliskan angka tersebut di lain tempat, (d) ulangi langkah 2 dan langkah 3 sampai semua angka sudah tercoret, (e) urutan angka yang dituliskan pada langkah 3 adalah permutasi acak dari angka awal. Algoritma Fisher-Yates terdiri dari dua metode, yakni metode *original* dan metode *modern*. Pada metode *modern*, angka yang terpilih tidak dicoret, tetapi posisinya ditukar dengan angka terakhir dari angka yang belum terpilih. Berikut ini adalah contoh pengerjaan dari versi *modern*: [12]

Tabel 1. Contoh Pengerjaan Algoritma Fisher-Yates

<i>Range</i>	<i>Roll</i>	<i>Scratch</i>	<i>Result</i>
		1 2 3 4 5 6 7 8	
1-8	6	1 2 3 4 5 8 7	6
1-7	2	1 7 3 4 5 8	2 6
1-6	6	1 7 3 4 5	8 2 6
1-5	1	5 7 3 4	1 8 2 6
1-4	3	5 7 4	3 1 8 2 6
1-3	3	5 7	4 3 1 8 2 6
1-2	1	7	5 4 3 1 8 2 6
Hasil Pengacakan :			7 5 4 3 1 8 2 6

2.2.8 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah sebuah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). [5] UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [13]

2.2.9 Android

Android is the most popular mobile operating system, based on the Linux Kernel, primarily designed for touchscreen mobile devices at the time this book was written. [14] Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile devices*) yang terdiri dari sistem operasi, *middleware*,

dan aplikasi-aplikasi utama. Awalnya, Android dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. [15]

2.2.10 Firebase

Firestore is a project by Google designed to help developers build apps without worrying about backend infrastructure. It operates on the freemium model, based on the amount of requests your backend has to respond to as well as the amount of storage you need. [16] Firestore is a cloud-based backend-as-a-service (BaaS) service provided by Google that provides a structural way to save your data very efficiently and also retrieve it at much faster speeds. Firestore has many features inbuilt which are very useful in a complete app development cycle. What makes Firestore handy is that it does not need any server side configurations. [17]

2.2.11 Android Studio

Android Studio is an integrated development environment (IDE), which is a very popular tool for Java development. [18] Android Studio is the best IDE for developing Android apps, and it is available for free to anyone who wants to develop professional Android apps. [19]

3. HASIL DAN PEMBAHASAN

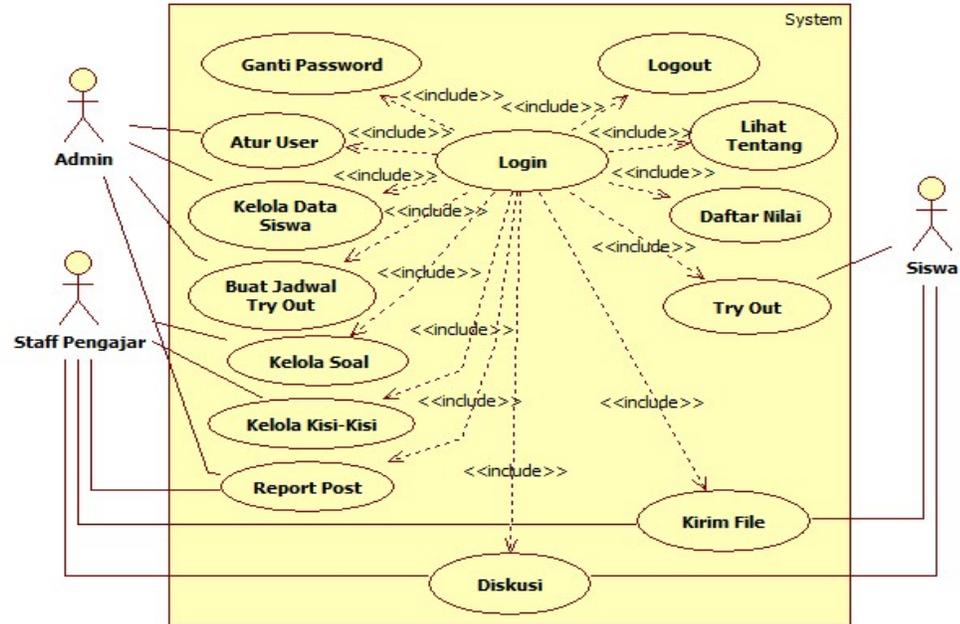
3.1 Perancangan Aplikasi Try Out Ujian Nasional

Aplikasi *try out* ini merupakan aplikasi yang sepenuhnya berbasis android. Aplikasi *try out* ini menggunakan algoritma Fisher-Yates dalam proses pengacakan soal sehingga setiap siswa mendapatkan soal yang berbeda satu sama lain. Dengan demikian, hal ini dapat mengurangi tingkat kecurangan siswa seperti menyontek saat kegiatan *try out* berlangsung. Berikut ini adalah fitur utama yang terdapat pada aplikasi *try out*:

- a. **Mengelola Data Siswa**
Pada bagian ini, admin dapat menambah, mengubah, dan menghapus data siswa kelas XII yang akan mengikuti kegiatan *try out*. Apabila data siswa telah dimasukkan, maka siswa tersebut dapat melakukan *login* di aplikasi menggunakan *username* dan *password* berupa nomor induk siswa.
- b. **Pembuatan Jadwal Try Out**
Pada bagian ini, admin dapat membuat atau mengubah jadwal pelaksanaan kegiatan *try out*. Jika jadwal telah dibuat, maka staff pengajar dapat memasukkan soal-soal yang akan diujikan pada kegiatan *try out* tersebut.
- c. **Daftar Nilai**
Pada bagian ini, pengguna aplikasi yang meliputi admin, siswa, dan staff pengajar dapat melihat daftar nilai dari *try out* yang telah dilaksanakan. Siswa hanya dapat melihat nilai *try out* yang diperoleh dirinya sendiri sedangkan admin dan staff pengajar dapat melihat seluruh nilai *try out* dari siswa.
- d. **Mengelola Kisi-Kisi**
Pada bagian ini, staff pengajar dapat menambah, mengubah, dan menghapus kisi-kisi dari mata pelajaran yang diajarkannya. Kisi-kisi ini berguna sebagai pedoman dalam menyusun soal yang akan diujikan saat pelaksanaan kegiatan *try out*. Dengan adanya kisi-kisi ini, setiap siswa dapat memperoleh soal *try out* dengan jumlah komposisi materi yang sama.
- e. **Mengelola Soal**
Pada bagian ini, staff pengajar dapat menambah, mengubah, dan menghapus soal yang akan diujikan saat pelaksanaan kegiatan *try out*. Proses memasukkan soal dapat dilakukan secara manual melalui aplikasi atau meng-*import file excel*. Jika soal yang dimasukkan semakin banyak, maka setiap siswa akan mendapatkan jenis soal yang semakin beragam pula satu sama lain.
- f. **Kirim File**
Pada bagian ini, staff pengajar dapat meng-*upload file* soal *try out* atau jawaban yang berjenis *file .pdf*. Siswa dapat men-*download file* tersebut sebagai bahan belajar untuk persiapan menghadapi ujian nasional.
- g. **Diskusi**
Pada bagian ini, staff pengajar dan sesama siswa dapat saling berdiskusi apabila siswa merasa kurang jelas dari pembahasan soal *try out* yang dilakukan saat di kelas. Siswa dapat melakukan *posting* pertanyaan dalam ruang diskusi ini, kemudian staff pengajar ataupun sesama siswa lain dapat memberikan komentar atau jawaban dari hasil *posting-an* tersebut. Namun, apabila siswa membuat *posting* atau komentar di luar materi pembelajaran, maka admin dapat memblokir akun siswa tersebut sehingga tidak dapat mengakses menu diskusi ini.
- h. **Report Post**
Pada bagian ini, staff pengajar dapat melakukan *report* pada *posting-an* siswa yang tidak sesuai dengan materi pembelajaran. Kemudian, admin dapat melihat *posting-an* yang telah di-*report* oleh staff pengajar sehingga admin dapat memblokir akun siswa tersebut.
- i. **Try Out**
Pada bagian ini, siswa melakukan kegiatan *try out* dengan cara menjawab soal-soal yang diambil secara acak dari *database*. Apabila siswa telah selesai mengerjakan soal atau waktu pengerjaan telah habis, maka jawaban akan dicocokkan oleh sistem dan nilai dari *try out* akan ditampilkan ke siswa serta tersimpan ke dalam *database*.

3.2 Perancangan Unified Modeling Language (UML)

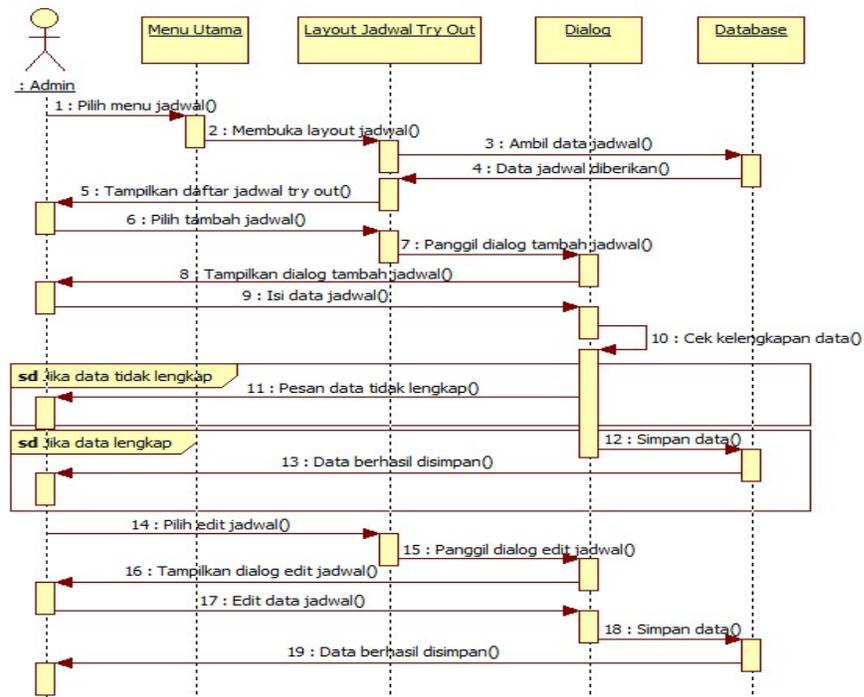
3.2.1 Diagram Use Case



Gambar 1. Diagram Use Case

Pengguna aplikasi dibagi menjadi tiga aktor yaitu admin, staff pengajar, dan siswa. Ketika membuka aplikasi, pengguna diwajibkan untuk melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password* supaya dapat menggunakan fitur yang ada pada aplikasi. Jika pengguna *login* dengan hak akses sebagai admin, maka pengguna dapat mengelola data siswa, membuat jadwal *try out*, mengatur *user*, ganti *password*, melihat daftar nilai, melihat *report post*, melihat tentang, dan *logout*. Sedangkan jika pengguna *login* dengan hak akses sebagai staff pengajar, maka pengguna dapat mengelola kisi-kisi, mengelola soal, melakukan diskusi dengan siswa, melakukan *report posting*-an siswa, melihat daftar nilai, mengirim *file* soal atau jawaban *try out*, ganti *password*, melihat tentang, dan *logout*. Apabila pengguna *login* dengan hak akses sebagai siswa, maka pengguna dapat melakukan *try out*, men-download *file* soal yang di-upload oleh staff pengajar, melakukan diskusi dengan staff pengajar, ganti *password*, melihat daftar nilai, melihat tentang, dan *logout*.

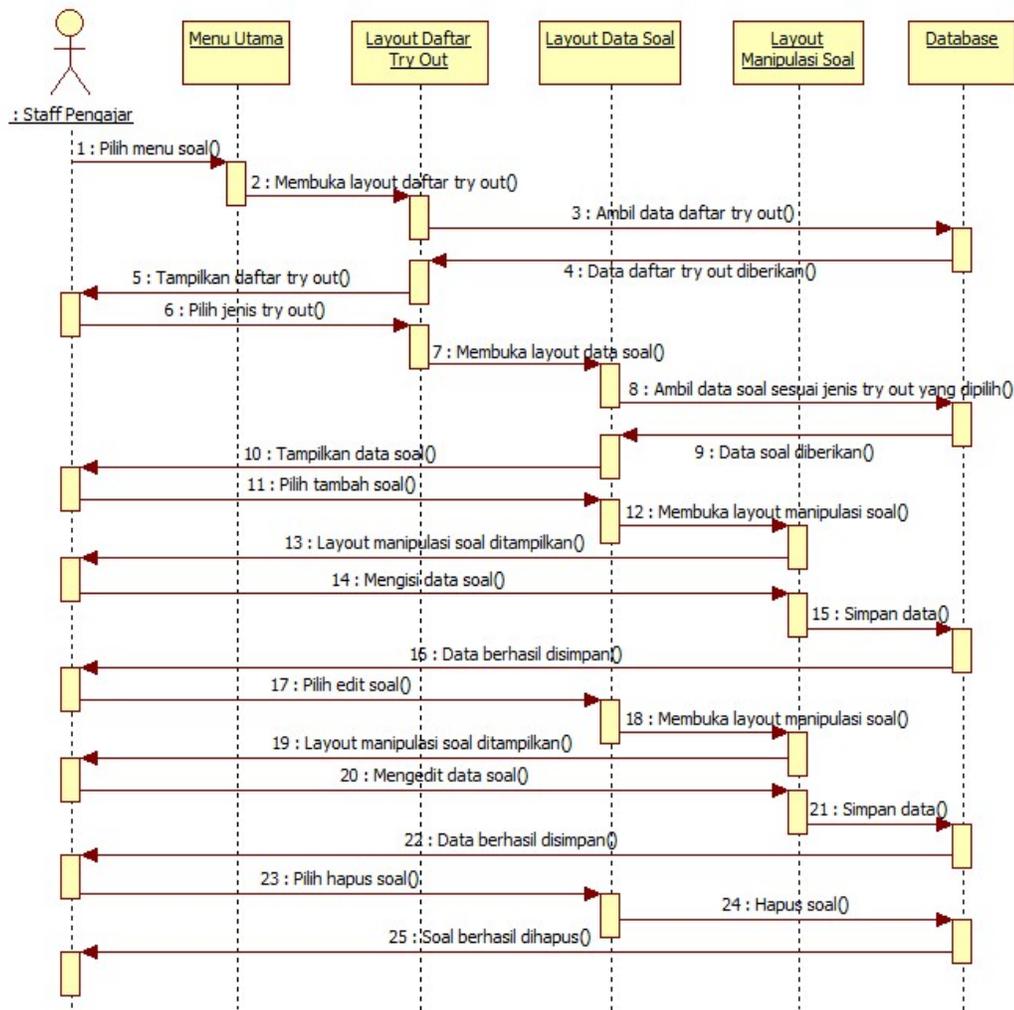
3.2.2 Diagram Sekuensial Buat Jadwal Try Out



Gambar 2. Diagram Sekuensial Buat Jadwal Try Out

Pada diagram tersebut, admin memilih menu jadwal *try out* dari menu utama. Kemudian, sistem akan menampilkan seluruh daftar jadwal *try out* yang tersimpan di *database*. Admin dapat membuat atau mengubah jadwal dengan mengisi data jadwal pada dialog isian yang muncul. Jika isian belum lengkap, maka sistem akan menampilkan pesan bahwa data belum lengkap. Jika isian lengkap, maka data jadwal *try out* akan disimpan ke *database*.

3.2.3 Diagram Sekuensial Kelola Data Soal

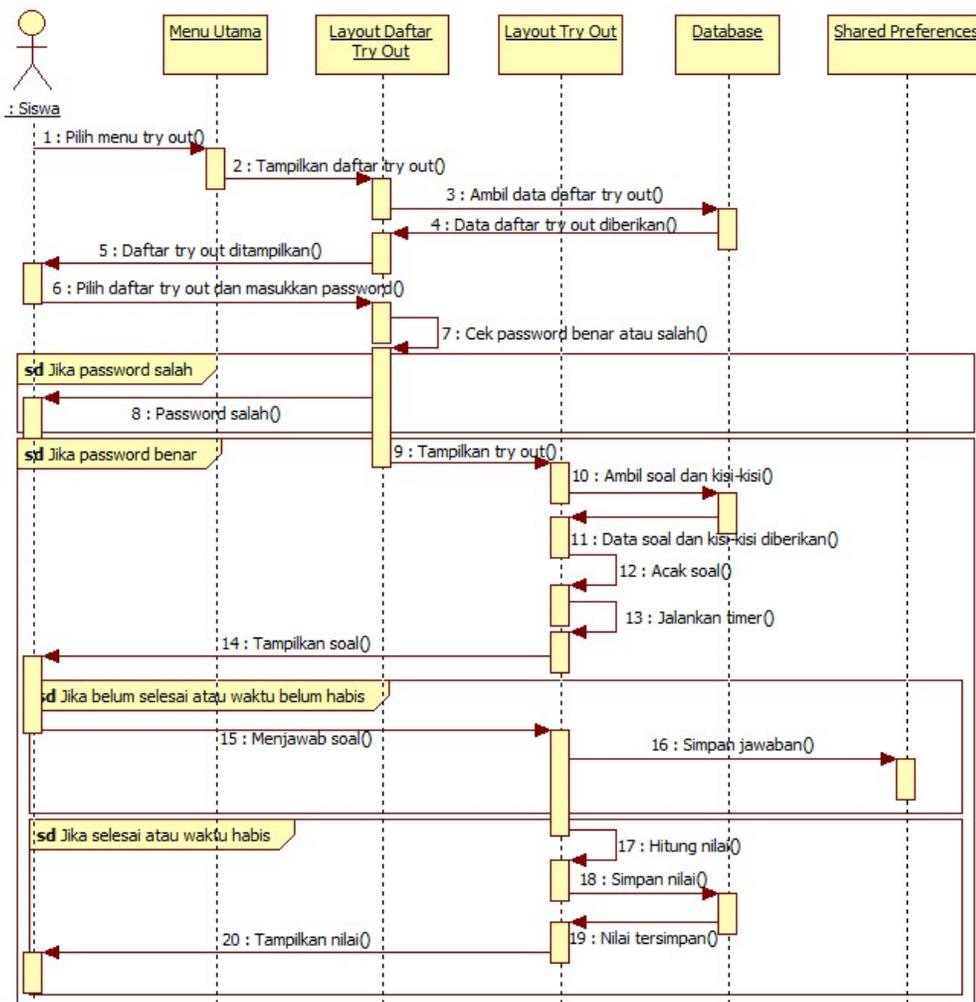


Gambar 3. Diagram Sekuensial Kelola Data Soal

Pada diagram tersebut, staff pengajar memilih menu soal dari menu utama. Kemudian, sistem akan menampilkan daftar jadwal *try out* yang telah dibuat oleh admin serta telah tersimpan di *database*. Staff pengajar dapat memilih salah satu dari daftar jadwal *try out* tersebut. Setelah itu, seluruh soal yang ada sesuai dengan jenis *try out* yang dipilih akan ditampilkan oleh sistem. Jika staff pengajar memilih menambah atau mengubah soal, maka sistem akan menampilkan *layout* manipulasi soal yang harus dilengkapi oleh staff pengajar. Data soal yang telah di-*update* akan tersimpan ke *database*. Selain itu, staff pengajar juga dapat menghapus soal yang telah tersimpan di *database*.

3.2.4 Diagram Sekuensial Try Out

Pada diagram tersebut, siswa memilih menu *try out* dari menu utama. Kemudian, sistem akan menampilkan daftar *try out* yang telah tersimpan di *database*. Siswa memilih daftar *try out* dan memasukkan *password* dari *try out* tersebut. Sistem akan mengecek *password* yang dimasukkan benar atau salah. Jika *password* salah, maka sistem menampilkan pesan bahwa *password* yang dimasukkan salah. Jika *password* benar, maka *layout try out* akan ditampilkan. Kemudian, sistem mengambil data soal dan kisi-kisi dari *database*. Setelah itu, sistem melakukan pengacakan soal, menjalankan *timer*, dan menampilkan soal-soal *try out*. Apabila siswa belum memilih selesai atau waktu belum habis, maka siswa dapat menjawab soal berpilihan ganda dan jawaban akan tersimpan ke *shared preferences*. Apabila siswa memilih selesai atau waktu telah habis, maka sistem melakukan perhitungan nilai *try out* yang didapatkan oleh siswa. Nilai tersebut akan tersimpan di *database* dan ditampilkan kepada siswa.

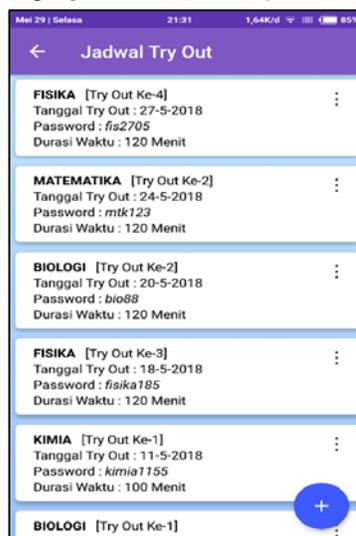


Gambar 4. Diagram Sekuensial Try Out

3.3 Tampilan Aplikasi

3.3.1 Tampilan Jadwal Try Out

Menu jadwal *try out* menampilkan seluruh jadwal *try out* yang akan dilaksanakan. Admin dapat membuat jadwal *try out* baru atau mengubah jadwal *try out* yang telah ada sedangkan staff pengajar dapat mengklik salah satu jadwal tersebut untuk meng-*input*-kan soal-soal *try out*.



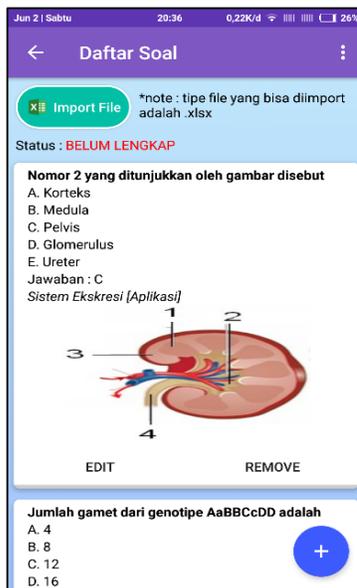
Gambar 5. Tampilan Jadwal Try Out

Berikut ini merupakan atribut yang terdapat pada tampilan jadwal *try out*:

- Tombol “+”, berfungsi untuk menampilkan kotak *dialog* penambahan jadwal *try out* baru ke *database*.
- Menu *pop-up* pada *card-view*, berfungsi untuk menampilkan kotak *dialog* dari jadwal *try out* yang akan di-*edit*.

3.3.2 Tampilan Daftar Soal

Setelah admin membuat jadwal *try out* yang akan dilaksanakan, staff pengajar baru dapat mengelola soal-soal untuk kegiatan *try out* tersebut. *Layout* daftar soal menampilkan seluruh soal yang telah tersimpan di *database*. Staff pengajar dapat menambah, mengubah, dan menghapus soal-soal *try out*. Pada *layout* daftar soal, terdapat menu *options* komposisi soal untuk menunjukkan jumlah soal pada setiap kisi-kisi.



Gambar 6. Tampilan Daftar Soal

Berikut ini adalah atribut yang terdapat pada tampilan daftar soal:

- Tombol “+”, berfungsi untuk menampilkan *layout* manipulasi data soal dalam keadaan isian kosong.
- Tombol *edit*, berfungsi untuk menampilkan *layout* manipulasi data soal dalam keadaan terisi sesuai soal *try out* yang dipilih untuk di-*edit*.
- Tombol *remove*, berfungsi untuk menghapus soal *try out* dari *database*.
- Tombol *import file*, berfungsi untuk meng-*import* soal *try out* melalui *file excel* yang berekstensi *.xlsx*.
- Textview* status, berfungsi untuk menunjukkan bahwa status soal *try out* belum lengkap atau sudah lengkap berdasarkan jumlah soal pada data kisi-kisi. Jika status soal belum lengkap, maka siswa tidak dapat melakukan kegiatan *try out*.

3.3.3 Tampilan Try Out



Gambar 7. Tampilan Try Out

Setelah siswa berhasil memasukkan *password* soal dengan benar, maka *layout try out* akan ditampilkan. Pada *layout try out*, siswa dapat menjawab soal-soal *try out* yang telah diambil secara acak dari *database* dengan cara memilih salah satu pilihan ganda. Waktu pengerjaan *try out* mulai dihitung mundur ketika soal telah ditampilkan ke siswa. Siswa dapat melakukan navigasi soal dari tombol yang telah disediakan. Jika pada soal *try out* terdapat gambar, maka gambar akan tampil di atas pertanyaan soal.

Berikut ini adalah atribut yang terdapat pada tampilan *try out*:

- Checkbox* ragu-ragu, berfungsi untuk menandai soal yang ragu-ragu jawabannya. Jika *checkbox* tercentang, maka warna tombol nomor pada daftar soal menjadi warna kuning.
- Radiobutton* pilihan ganda, berfungsi untuk memilih jawaban dari soal *try out*. Jika salah satu *radiobutton* terpilih, maka warna tombol nomor pada daftar soal menjadi warna hijau.
- Tombol *prev*, berfungsi untuk menampilkan soal pada nomor sebelumnya.
- Tombol *next*, berfungsi untuk menampilkan soal pada nomor selanjutnya. Jika berada pada soal terakhir, maka tombol *next* akan berubah menjadi tombol *finish*.
- Tombol daftar soal, berfungsi untuk menampilkan kotak *dialog* nomor soal.



Gambar 8. Tampilan Dialog Nilai Try Out

Setelah siswa mengakhiri kegiatan *try out* atau waktu pengerjaan telah habis, maka *dialog* nilai *try out* akan ditampilkan ke siswa untuk memberitahukan nilai yang didapatkan oleh siswa. Kemudian, nilai *try out* tersebut akan tersimpan ke *database*.

3.4 Penerapan Algoritma Fisher-Yates

Algoritma Fisher-Yates merupakan sebuah algoritma yang menghasilkan permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Pada gambar 9, terdapat contoh *list* soal *try out* sebelum penerapan algoritma Fisher-Yates. Urutan soal-soal *try out* tersebut sesuai dengan urutan di *database* ketika soal dimasukkan oleh staff pengajar.

[0] Jika diketahui $K_a \text{CH}_3\text{COOH} = 10^{-5}$, maka pH larutan $\text{Ca}(\text{CH}_3\text{COO})_2$ 0,1 M adalah
[1] Jenis hidrolisis yang terjadi dari senyawa HCN dengan NH_3 adalah hidrolisis
[2] Proses cuci darah merupakan penerapan dari sifat koloid
[3] Terbentuknya delta di muara sungai merupakan sifat koloid
[4] Jika 200 ml larutan H_2SO_4 0,5 M dititrasi dengan 200 ml larutan NaOH, maka konsentrasi larutan NaOH adalah
[5] Indikator untuk mengetahui titik akhir titrasi adalah
[6] Bilangan oksidasi atom S pada senyawa H_2SO_4 adalah
[7] Bilangan oksidasi +3 dari unsur Cl terdapat pada senyawa
[8] Pada penyepuhan perak dengan emas, logam yang dipakai sebagai katode adalah
[9] Zat yang terdapat pada katoda dari elektrolisis larutan KI adalah

Gambar 9. List Soal Sebelum Penerapan Algoritma Fisher-Yates

Ketika algoritma Fisher-Yates dijalankan, maka akan terjadi proses perpindahan posisi soal sesuai *index* yang dimiliki selama proses pengulangan. Proses pengulangan dilakukan sebanyak jumlah soal yang terdapat pada *arraylist*. Adapun tahapan proses algoritma Fisher-Yates saat melakukan pengulangan untuk mengubah posisi *index* soal dapat dilihat pada tabel berikut ini:

Tabel 2. Perubahan Posisi Index Soal

i	indexRand	listAllSoal	Hasil
		0 1 2 3 4 5 6 7 8 9	
9	1	0 9 2 3 4 5 6 7 8	1
8	7	0 9 2 3 4 5 6 8	7 1
7	4	0 9 2 3 8 5 6	4 7 1
6	6	0 9 2 3 8 5	6 4 7 1
5	3	0 9 2 5 8	3 6 4 7 1
4	0	8 9 2 5	0 3 6 4 7 1
3	1	8 5 2	9 0 3 6 4 7 1
2	1	8 2	5 9 0 3 6 4 7 1
1	1	8	2 5 9 0 3 6 4 7 1
Hasil Pengacakan :			8 2 5 9 0 3 6 4 7 1

Tabel 2 menunjukkan posisi *index* dari soal yang awalnya berurutan mengalami perubahan secara bertahap sampai proses pengulangan selesai dilakukan. Dengan demikian, dapat dilihat pada gambar 10 yang menunjukkan urutan soal pada *arraylist* setelah algoritma Fisher-Yates diterapkan mengalami perubahan yang cukup signifikan dibandingkan dengan keadaan awal yang terdapat pada gambar 9. Hal ini telah menunjukkan bahwa algoritma Fisher-Yates telah berhasil mengacak kumpulan soal yang awalnya berurutan menjadi tidak berurutan.

[8] Pada penyepuhan perak dengan emas, logam yang dipakai sebagai katode adalah
[2] Proses cuci darah merupakan penerapan dari sifat koloid
[5] Indikator untuk mengetahui titik akhir titrasi adalah
[9] Zat yang terdapat pada katoda dari elektrolisis larutan KI adalah
[0] Jika diketahui $K_a \text{CH}_3\text{COOH} = 10^{-5}$, maka pH larutan $\text{Ca}(\text{CH}_3\text{COO})_2$ 0,1 M adalah
[3] Terbentuknya delta di muara sungai merupakan sifat koloid
[6] Bilangan oksidasi atom S pada senyawa H_2SO_4 adalah
[4] Jika 200 ml larutan H_2SO_4 0,5 M dititrasi dengan 200 ml larutan NaOH, maka konsentrasi larutan NaOH adalah
[7] Bilangan oksidasi +3 dari unsur Cl terdapat pada senyawa
[1] Jenis hidrolisis yang terjadi dari senyawa HCN dengan NH_3 adalah hidrolisis

Gambar 10. List Soal Setelah Penerapan Algoritma Fisher-Yates

4. KESIMPULAN

Berdasarkan perancangan dan implementasi aplikasi *try out* ujian nasional berbasis Android yang telah dilakukan, maka penulis dapat menarik beberapa kesimpulan sebagai berikut:

- Aplikasi *try out* ujian nasional dapat mengatasi kelemahan yang ada pada pelaksanaan *try out* secara manual di Bimbingan Belajar Anastasia seperti mengurangi penggunaan kertas, menghemat waktu dalam pengoreksian hasil *try out*, dan mengurangi tingkat kecurangan siswa.
- Penerapan algoritma Fisher-Yates yang digunakan pada aplikasi *try out* ujian nasional dapat mengacak soal-soal *try out* sehingga setiap siswa memperoleh soal yang berbeda satu sama lain.
- Jika jumlah soal yang dimasukkan banyak untuk setiap materi, maka soal yang didapatkan oleh setiap siswa akan semakin beragam.
- Aplikasi *try out* ujian nasional ini dapat memberikan persiapan akademik bagi siswa yang akan mengikuti ujian nasional.
- Aplikasi *try out* ujian nasional hanya dapat dijalankan dengan dukungan koneksi internet.

5. SARAN

Setelah melakukan implementasi aplikasi *try out* ujian nasional dengan algoritma Fisher-Yates pada Bimbingan Belajar Anastasia berbasis Android, penulis menyadari bahwa aplikasi yang dihasilkan belum sempurna. Untuk itu, ada beberapa saran agar aplikasi *try out* ini dapat dikembangkan dan digunakan sebagai acuan untuk menghasilkan aplikasi yang lebih baik lagi, antara lain:

- a. Menambah jenis soal *try out* seperti soal isian atau uraian sehingga tidak hanya terbatas pada soal pilihan ganda.
- b. Algoritma Fisher-Yates dapat dikombinasikan dengan algoritma lain untuk menghasilkan soal-soal *try out* yang lebih variatif.
- c. Mengembangkan aplikasi menjadi berbasis web bagi admin dan staff pengajar sehingga proses peng-input-an data menjadi lebih cepat.
- d. Menambahkan fitur *group chat* serta *video call* di ruang diskusi.

UCAPAN TERIMA KASIH

Dalam penulisan ini, penulis telah banyak mendapatkan bantuan berupa bimbingan, petunjuk, data, saran maupun dorongan moral dari berbagai pihak, maka pada kesempatan ini penulis mengucapkan terima kasih kepada civitas akademika STMIK Widya Dharma Pontianak dan kepada Ibu Ir. Anastasia selaku pemilik dari Bimbingan Belajar Anastasia yang telah memberi kesempatan kepada penulis untuk melakukan penelitian yang berkenaan dengan penulisan tugas akhir ini.

DAFTAR PUSTAKA

- [1] Sutabri, Tata. (2014). *Pengantar Teknologi Informasi*. Andi. Yogyakarta.
- [2] Setiawan, Rudy. (2017). *Sistem Operasi*. Seribu Bintang. Jawa Timur.
- [3] Hakim, Rachmad. (2010). *Buku Pintar Windows 7*. PT Elex Media Komputindo. Jakarta.
- [4] Irwansyah, Edi dan Jurike V. Moniaga. (2014). *Pengantar Teknologi Informasi*. Deepublish. Yogyakarta.
- [5] Nugroho, Adi. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Andi. Yogyakarta.
- [6] Sutabri, Tata. (2016). *Sistem Informasi Manajemen*. Andi. Yogyakarta.
- [7] Rizky, Soetam. (2011). *Konsep Dasar Rekayasa Perangkat Lunak*. Prestasi Pustaka. Jakarta.
- [8] Yasin, Verdi. (2012). *Rekayasa Perangkat Lunak Berorientasi Objek-Pemodelan, Arsitektur, dan Perancangan (Modelling, Architecture and Design)*. Mitra Wacana Media. Jakarta.
- [9] Elfiza, Yulia, Rusman, dan M. Nasir. (2016). Hubungan antara Hasil Uji Kognitif Try Out Ujian Nasional (UN) dengan Hasil Ujian Nasional (UN) Mata Pelajaran Kimia SMA Kota Banda Aceh Tahun Ajaran 2014/2015. *Jurnal Ilmiah Mahasiswa Pendidikan Kimia*. vol. 1, no. 3: hal. 35-42.
- [10] Harmadya, Sasmita, dan Wirdiani. (2015). Rancang Bangun Aplikasi Tryout Ujian Nasional Sekolah Menengah Pertama (Smp) Berbasis Android. *Jurnal Lontar Komputer*. vol. 6, no. 2:hal. 108-119.
- [11] Ekojono, Dyah Ayu Irawati, dan Lugman Affandi. (2017). Penerapan Algoritma Fisher-Yates pada Pengacakan Soal Game Aritmatika. *Prosiding Sentia*. vol. 9, no. 1: hal. 95-100.
- [12] Hadi, Ahmaddul. (2014). Pengembangan Sistem Informasi Ujian Online Berbasis Web dengan Pengacakan Soal Menggunakan Algoritma Fisher-Yates Shuffle. *Jurnal Teknologi Informasi & Pendidikan*. vol. 7, no. 2:hal. 161-170.
- [13] Sukamto, Rosa A. dan M. Shalahuddin. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika. Bandung.
- [14] Dunham, Ken., et al. (2015). *Android Malware and Analysis*. CRC Press. New York.
- [15] Juhara, Zamrony P. (2016). *Panduan Lengkap Pemrograman Android*. Andi. Yogyakarta.
- [16] Domes, Scott. (2017). *Progressive Web Apps with React*. Packt Publishing Ltd. Birmingham.
- [17] Mainkar, Prajyot. (2017). *Expert Android Programming*. Packt Publishing Ltd. Birmingham.
- [18] Annuzi, Joseph Jr., Lauren Darcey, Shane Conder. (2014). *Advanced Android Application Development*. Addison-Wesley. United States.
- [19] Drongelen, Mike Van. (2015). *Android Studio Cookbook*. Packt Publishing Ltd. Birmingham.