

Rancang Bangun Media Pembelajaran *Tenses English* Berbasis Android menggunakan Algoritma *Fisher-Yates*

Kiki Vebiant¹, M. Iwan Wahyuddin², Ratih Titi Komala Sari³

^{1,2,3} Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional

article info

Article history:

Received 15 December 2020

Received in revised form

13 January 2021

Accepted 17 January 2021

Available *online* October 2021

DOI:

<https://doi.org/10.35870/jtik.v5i4.246>

Keywords:

Android, *Fisher-Yates*, *Tenses*, Learning Application.

Kata Kunci:

Android, *Fisher-Yates*, *Tenses*, Aplikasi Pembelajaran.

abstract

English is a language that is widely used in various countries in the world. This makes english an international language and also must be studied in schools or even universities. In english, the verb used will be different at different times, places, and also events or it can be called a change in the verb (tense). So, learning and mastering tenses in english is the basis for learning or mastering english. This study aims to design and build an Android-based learning media for english tenses. The design applies a form of english tense quiz using the Fisher-Yates algorithm, which is a randomization algorithm that can randomize the order of the quiz questions so that the order of the quiz questions is not always the same and there are no repeated quizzes. The expected results will later become a learning medium for people who want to learn or master english and can understand the tenses in english.

abstrak

Bahasa inggris merupakan bahasa yang banyak digunakan pada berbagai negara-negara didunia. Hal tersebut menjadikan bahasa Inggris menjadi internasional dan juga harus dipelajari pada sekolah maupun pada perguruan tinggi. Didalam bahasa inggris, kata kerja yang digunakan akan berbeda pada waktu, tempat, maupun kejadian yang berbeda atau bisa disebut dengan perubahan kata kerja (tense). Sehingga, mempelajari dan menguasai tenses dalam bahasa inggris merupakan dasar dalam mempelajari ataupun menguasai bahasa inggris. Penelitian ini bertujuan merancang dan membangun media pembelajaran tenses bahasa inggris berbasis Android. Perancangan menerapkan bentuk kuis tenses bahasa inggris menggunakan algoritma Fisher-Yates, yaitu algoritma pengacakan yang dapat mengacak urutan soal kuis sehingga urutan soal kuis tidak selalu sama dan juga tidak adanya kuis yang terulang. Hasil yang diharapkan nantinya dapat menjadi media pembelajaran untuk masyarakat yang ingin belajar ataupun menguasai bahasa Inggris dapat memahami tenses dalam bahasa inggris.

*Corresponding author. Email: kvebiant@gmail.com¹, iwan.wahyuddi@civitas.unas.ac.id², ratih.titi@civitas.unas.ac.id³.

© E-ISSN: 2580-1643.

Copyright © 2021. Published by Lembaga Informasi dan Riset (KITA INFO dan RISET), Lembaga KITA (<http://creativecommons.org/licenses/by/4.0/>).

1. Latar Belakang

Bahasa Inggris merupakan bahasa Internasional yang banyak digunakan pada berbagai negara-negara didunia. Hal ini menjadikan bahasa Inggris menjadi bahasa yang harus dipelajari pada sekolah dasar maupun sampai pada perguruan tinggi. Banyak sekali dasar-dasar untuk mempelajari ataupun menguasai bahasa Inggris, termasuk mempelajari *tenses* atau perubahan kata kerja pada waktu yang berbeda. *Tenses* dapat mempengaruhi makna dari sebuah kalimat dalam bahas Inggris, maka dari itu penggunaan *tenses* dapat berpengaruh signifikan bagi penerima informasi atau komunikasi.

Dengan demikian media pembelajaran *tenses* dapat membantu masyarakat yang ingin mempelajari atau menguasai bahasa Inggris. Metode pembelajaran *tenses* dapat menggunakan kuis atau soal untuk melatih *tenses* bahasa inggris pengguna, maka itu diperlukan fungsi untuk melakukan pengacakan kuis atau soal. Pengacakan kuis yang dipilih adalah algoritma *Fisher-Yates* karena metode pengacakan algoritmanya optimal.

Dari penelitian terdahulu, dilakukan penelitian untuk pengacakan soal menggunakan algoritma *Fisher-Yates* pada tampilan setiap peserta ujian agar berbeda sehingga peserta memiliki nomor soal yang sama tetapi bentuk soal yang saling berbeda [1]. Adapula penelitian lain yang memanfaatkan metode *Fisher-Yates* untuk digunakan dalam pengacakan kuis atau soal fiqih pada aplikasi *mobile learning*, dimana ditunjukkan setiap pengujiannya memiliki jenis soal yang berbeda disetiap nomor soal [2].

Kemudian dalam penelitian lain, dilakukan implementasi algoritma *Fisher-Yates* pada kuis *game* sehingga pertanyaan atau soal kuis teracak sendiri sehingga membuat permainan tidak mudah ditebak saat mengulang kembali permainan tersebut [3]. Penelitian lain juga memanfaatkan algoritma *Fisher-Yates* untuk pengacakan soal ujian masuk POLRI yang membuat hasil dari pengujian sistem sesuai harapan [4].

Adapula penelitian serupa dari Iyon Maryono dkk, dimana dalam penelitiannya memanfaatkan Algoritma *Fisher-Yates* untuk pengacakan pada kuis aljabar pada aplikasi hybrid yang menunjukkan

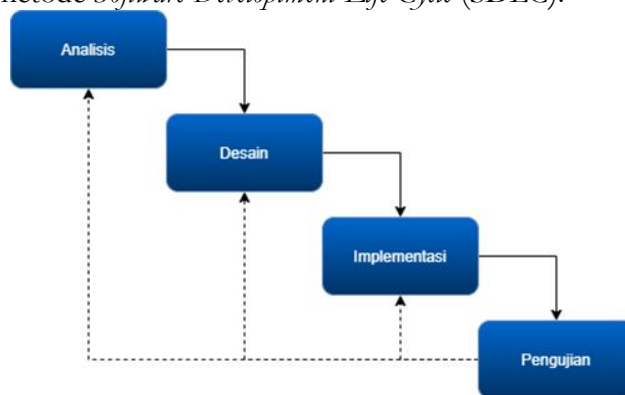
bahwa algoritma *Fisher-Yates* sangatlah cocok untuk metode pengacakan kuis [5]. Kemudian juga terdapat penelitian serupa pada aplikasi multimedia interaktif untuk pembelajaran *tenses* pada bahasa Inggris yang dimana menunjukkan urutan soal tidak dapat ditebak karena algoritma *Fisher-Yates* menggunakan math random untuk mengubah urutan *array* yang dipanggil [6].

Penelitian lain juga menggunakan algoritma *Fisher-Yates* untuk mengacak urutan soal kuis pada sistem penerimaan mahasiswa baru di STMIK-AMIK RIAU dan juga menggunakan metode SDLC atau bisa disebut dengan *System Development Life Cycle* [7]. Penelitian lain juga dilakukan oleh Marlinda dkk, yang dimana penelitian tersebut mengacak soal *try out* ujian semester dengan menggunakan metode SDLC [8].

Berdasarkan penelitian terdahulu penggunaan algoritma *Fisher-Yates* merupakan algoritma yang cocok pada media pembelajaran berbasis kuis. Maka penelitian kali ini akan menggunakan algoritma *Fisher-Yates* untuk media pembelajaran *tenses* pada bahasa Inggris berbasis Android.

2. Metodologi Penelitian

Tahapan metode penelitian yaitu akan menggunakan metode *Software Development Life Cycle* (SDLC).



Gambar 1. Metode SDLC

Gambar 1 adalah metode SDLC yang memiliki 4 tahapan yaitu tahapan analisis, desain, implementasi dan pengujian.

1) Analisis

Tahap analisis pada metode SDLC disini adalah proses pengumpulan data untuk menentukan dan mengetahui metode yang digunakan, kebutuhan perangkat, pembuatan *flowchart* pada sistem,

spesifikasi yang dibutuhkan untuk aplikasi yang akan dibuat.

2) Desain

Tahap desain pada metode SDLC disini adalah proses pembuatan interface aplikasi, konsep aplikasi dan juga pembuatan fitur aplikasi.

3) Implementasi

Tahap implementasi pada metode SDLC disini adalah melakukan penerapan *interface* yang telah dibuat kepada sistem dan menjalankan aplikasi dengan metode yang digunakan.

4) Pengujian

Tahap pengujian pada metode SDLC disini adalah proses pemeriksaan aplikasi yang telah dibuat yaitu dengan cara menentukan tujuan aplikasi yang telah ditetapkan pada penelitian sudah terpenuhi.

Algoritma Fisher-Yates Shuffle

Algoritma *Fisher-Yates shuffle* adalah algoritma yang menghasilkan mutasi yang acak dari himpunan yang terbatas. Jika diterapkan dengan benar, maka hasil pengacakan dari algoritma *Fisher-Yates* memiliki probabilitas yang sama sehingga tidak adanya bias pada himpunan [9, 11].

Proses dari pengacakan algoritma *Fisher-Yates* hampir sama dengan mengambil kartu satu persatu dari tumpukan kartu sampai tidak ada yang tersisa [10], metode *Fisher-Yates* sangat dianggap optimal oleh banyak orang untuk menghasilkan permutasi yang acak dan juga benar [9]. Pengacakan algoritma *Fisher-Yates* merupakan metode yang sangat optimal dengan waktu yang dilihat dari *flowchart* algoritma, algoritma *Fisher-Yates* juga tidak memakai kapasitas ruang memori penyimpanan yang terlalu besar [12].

Algoritma *Fisher-Yates* pertama kali diusulkan pada tahun 1938 oleh Ronald Fisher dan Frank Yates yang dikaji pada tahun 1948 dengan versi yang modern sebagai sebuah varian [13]. Selain itu, algoritma ini banyak digunakan karena pengacakan data yang dihasilkan tidak akan sama. Jadi, ini menghindari pengulangan dan duplikasi [14].

Keuntungan lain dari *Fisher-Yates* adalah algoritma ini menghasilkan hasil yang tidak bias. Setiap permutasi sebuah array memiliki kemungkinan yang sama [15]. Algoritma *Fisher-Yates* termasuk dalam kategori algoritma dinamis dikarenakan hasil dari pengacakan

tidak ada yang bias bila dilakukan pengacakan kembali. Algoritma *Fisher-Yates* memiliki beberapa tahapan dalam proses pengacakan. Berikut adalah tahapan-tahapan dari proses pengacakan sebagai berikut [16, 17]:

Tabel 1. Proses pengacakan algoritma *Fisher-Yates*

| Jarak | Baris | Nilai | Hasil |
|-------|-------|----------------------|----------------------|
| | | 1,2,3,4,5,6,7,8,9,10 | |
| 1-10 | 5 | 1,2,3,4,6,7,8,9,10 | 5 |
| 1-9 | 7 | 1,2,3,4,6,7,9,10 | 8,5 |
| 1-8 | 3 | 1,2,4,6,7,9,10 | 3,8,5 |
| 1-7 | 5 | 1,2,4,6,9,10 | 7,3,8,5 |
| 1-6 | 4 | 1,2,4,9,10 | 6,7,3,8,5 |
| 1-5 | 2 | 1,4,9,10 | 2,6,7,3,8,5 |
| 1-4 | 1 | 4,9,10 | 1,2,6,7,3,8,5 |
| 1-3 | 3 | 4,9 | 10,1,2,6,7,3,8,5 |
| 1-2 | 2 | 4 | 9,10,1,2,6,7,3,8,5 |
| | | | 4,9,10,1,2,6,7,3,8,5 |

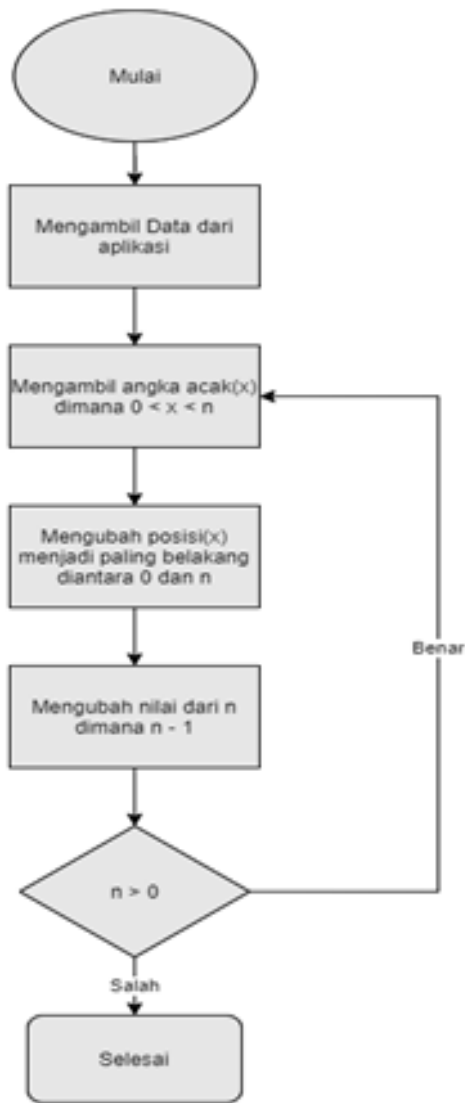
Tabel 1 menjelaskan alur dari proses pengacakan algoritma *Fisher-Yates* yang diantaranya:

- 1) Menentukan jumlah data yang ada.
- 2) Memilih data dari 1 sampai n yang belum dipindahkan secara acak.
- 3) Memindahkan data yang dipilih ke tempat lain.
- 4) Melakukan perulangan langkah 2 dan 3 sampai semua data habis atau $n = 0$.
- 5) Hasil dari data yang dipindah masuk ke kolom hasil yaitu adalah tempat lain dari data pengacakan tersebut.

Berdasarkan referensi-referensi yang telah dikumpulkan, maka penelitian ini menggunakan algoritma *Fisher-Yates* sebagai metode pengacakan soal atau kuis pada aplikasi *tenses* bahasa Inggris berbasis Android agar aplikasi dapat bekerja secara harapan.

Perancangan Sistem

Pada penelitian ini dibuatlah aplikasi khusus yang mendukung proses pembelajaran *tenses* bahasa Inggris pada aplikasi *mobile* android. Aplikasi *mobile* android yang dibuat menggunakan *framework flutter*, bahasa pemrograman dart, dan aplikasi Android Studio.

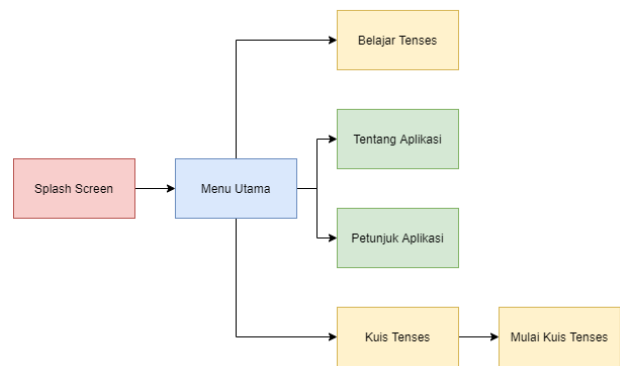


Gambar 2. Flowchart Sistem

Pada gambar 2 adalah *flowchart* dari aplikasi sistem pengacakan algoritma *Fisher-Yates* pada soal *tenses* bahasa Inggris. Tahap pertama yaitu sistem akan mengambil data semua soal yang ada pada aplikasi, lalu mengambil angka acak dari nilai 0 sampai pada nilai tertinggi soal atau soal paling terakhir. Setelah itu mengubah posisi angka yang sudah diacak yang sebelumnya untuk mengubah posisi angka soal yang sudah diacak menjadi soal paling belakang atau soal paling terakhir. Selanjutnya akan melakukan perulangan sampai semua soal sudah diacak.

User Flow Sistem

Pada tahapan ini menentukan user flow dari sistem, user flow digunakan untuk menggambarkan jalannya sistem aplikasi android yang dibuat pada penelitian ini dari awal hingga akhir aplikasi.



Gambar 3. User Flow Sistem

Pada gambar 3 adalah *user flow* atau alur pada aplikasi Android yang penggunaan aplikasinya dimulai dari halaman *splash screen*, menu utama, tentang aplikasi, petunjuk aplikasi, belajar *tenses* dan kuis *tenses*. *User flow* pada aplikasi mempunyai peran masing-masing, yaitu:

- 1) *Splash screen*: merupakan halaman tampilan awal dari aplikasi yang berfungsi untuk mengantarkan pengguna ke halaman menu utama.
- 2) Menu utama: merupakan halaman yang memiliki menu - menu dan halaman lain seperti tentang aplikasi, petunjuk aplikasi, belajar *tenses* dan kuis *tenses*.
- 3) Tentang aplikasi: merupakan tampilan yang berisikan informasi - informasi tentang aplikasi.
- 4) Petunjuk aplikasi: merupakan tampilan yang berisikan petunjuk-petunjuk untuk menggunakan aplikasi.
- 5) Belajar *tenses*: merupakan halaman tampilan yang berisikan pelajaran - pelajaran *tenses* bahasa Inggris dari awal sampai akhir.
- 6) Kuis *tenses*: merupakan halaman tampilan yang berisikan soal-soal pelajaran *tenses* bahasa Inggris yang dimana soal tersebut tidak akan sama urutannya pada saat mengulang kembali.

3. Hasil dan Pembahasan

Analisa Kebutuhan Sistem

Dalam tahapan analisa kebutuhan sistem bertujuan untuk mengidentifikasi kebutuhan dalam merancang sistem aplikasi yang akan dibuat pada penelitian ini. Didapatkan kebutuhan dari sistem yaitu berupa perangkat lunak (*software*) dan juga perangkat keras (*hardware*). Berikut adalah kebutuhan *software* dan *hardware* dalam penelitian ini:

Tabel 2. Kebutuhan Perangkat Keras

| Perangkat | Spesifikasi |
|----------------|---|
| Processor | Intel Core i5-5000 series atau lebih tinggi |
| Ram | 8GB atau lebih tinggi |
| Harddisk | 250GB atau lebih tinggi |
| Sistem Operasi | Windows 8/10 |
| Smartphone | Android |

Tabel 2 menunjukkan kebutuhan perangkat keras yang digunakan untuk menyelesaikan penelitian dan aplikasi android penelitian. Kebutuhannya adalah Komputer dengan spesifikasi processor Intel Core i5-500 atau lebih tinggi, ram yang memiliki kapasitas 8GB atau lebih tinggi, harddisk yang memiliki kapasitas 250GB atau lebih tinggi dan juga sistem operasi computer yaitu windows 8/10 dan juga smartphone android untuk pengujian aplikasi.

Tabel 3. Kebutuhan perangkat lunak

| Perangkat | Spesifikasi |
|----------------|------------------|
| Android Studio | IDE sistem |
| Flutter | Framework sistem |
| Dart | Bahasa sistem |

Tabel 3 menunjukkan kebutuhan perangkat lunak yang digunakan untuk menyelesaikan penelitian dan aplikasi android penelitian. Kebutuhannya adalah aplikasi IDE Android Studio agar dapat menggunakan framework flutter dan bahasa dart untuk membuat aplikasi android.

Tampilan Aplikasi

Dalam perancangan sistem aplikasi, peneliti menggunakan *framework flutter* untuk membuat desain, kode sistem aplikasi, logika dan lain sebagainya.



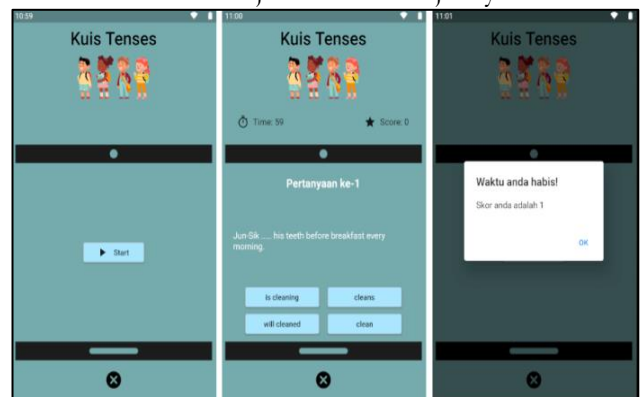
Gambar 4. Tampilan halaman menu utama

Gambar 4 merupakan tampilan halaman menu utama dari aplikasi yang terdiri dari 5 button yaitu button belajar *tenses* berfungsi untuk berpindah ke halaman belajar *tenses*, button kuis *tenses* berfungsi untuk berpindah ke halaman kuis *tenses*, button tentang aplikasi yang berfungsi untuk memunculkan informasi tentang aplikasi, button petunjuk yang berfungsi untuk memunculkan informasi petunjuk penggunaan aplikasi dan button silang untuk keluar dari aplikasi.



Gambar 5. Tampilan halaman Belajar Tenses

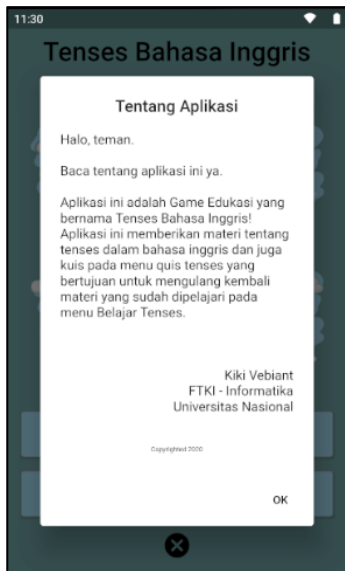
Gambar 5 merupakan tampilan halaman belajar materi *tenses* bahasa inggris dari aplikasi yang dimana terdapat penjelasan tentang *tenses* bahasa inggris dan juga contohnya. Terdapat juga 3 button yang memiliki fungsi berbeda yaitu button silang untuk kembali ke halaman menu utama, button panah kiri untuk kembali ke materi sebelumnya dan button panah kekanan untuk menuju materi selanjutnya.



Gambar 6. Tampilan halaman Kuis Tenses

Gambar 6 merupakan tampilan halaman kuis *tenses* yang dimana pengguna menjawab soal-soal yang sudah ditentukan tetapi soal selalu acak. Pada awal

halaman kuis ada 2 tombol yaitu tombol start untuk memulai soal dan tombol silang untuk 412embali, pada saat sudah mulai mengerjakan soal diberikan waktu 1 menit untuk mengerjakan soal dan ada 5 tombol yaitu 4 tombol jawaban dan 1 tombol silang untuk 412embali, selanjutnya setelah waktu telah habis akan menunjukkan informasi bahwa waktu telah habis dan berapa skor yang didapatkan lalu 412embali ke halaman awal kuis *tenses*.



Gambar 7. Tampilan tentang aplikasi

Pada gambar 7 adalah tampilan menu tentang aplikasi yang dimana terdapat informasi tentang aplikasi dan juga terdapat 1 buah tombol ok untuk kembali menuju halaman menu utama.



Gambar 8. Tampilan petunjuk

Gambar 8 adalah tampilan menu petunjuk yang

dimana terdapat informasi untuk menggunakan aplikasi, segala jenis fitur yang terdapat dalam aplikasi dan juga terdapat 1 buah tombol yaitu tombol OK untuk kembali ke halaman menu utama.

Pengujian

Pengujian pada penelitian ini dilakukan melalui beberapa tahap, diantaranya:

1) Pengujian Device

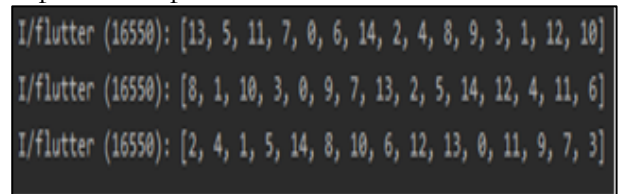
Pada tahap pengujian device yaitu proses menjalankan aplikasi pada beberapa device android yang memiliki sistem operasi yang berbeda agar mengetahui keberhasilan aplikasi pada setiap device android yang berbeda. Maka dari itu, pengujian ini dilakukan terhadap 5 device. Pengujian device dapat dilihat pada Tabel 4.

Tabel 4. Pengujian Device

| Perangkat | Sistem Operasi | Hasil |
|-----------|---------------------|----------|
| 1 | Android Q | Berhasil |
| 2 | Android Pie | Berhasil |
| 3 | Android Nougat | Berhasil |
| 4 | Android Marshmallow | Berhasil |
| 5 | Android Lollipop | Berhasil |

2) Pengujian Algoritma

Pada tahap pengujian algoritma di penelitian ini adalah pengujian algoritma *Fisher-Yates* untuk pengacakan soal kuis pada aplikasi. Pengujian algoritma dilakukan sebanyak 3 kali untuk memastikan tidak adanya pengulangan soal kuis. Pengujian algoritma *Fisher-Yates* pada aplikasi dapat dilihat pada Gambar 9.



Gambar 9. Tampilan pengujian algoritma

3) Pengujian Aplikasi

Pada tahap Pengujian aplikasi ini yaitu melakukan pengujian menggunakan metode blackbox testing untuk mengetahui fungsionalitas dari aplikasi tercapai atau tidak. Pengujian aplikasi menggunakan metode blackbox testing dapat dilihat pada Tabel 5.

Tabel 5. Pengujian aplikasi

| Pertanyaan | Pengujian | Hasil |
|----------------------|----------------------------------|----------|
| Gambar | Gambar tertampilkan dengan jelas | Berhasil |
| Tombol | Tombol responsif | Berhasil |
| Teks | Teks tertampilkan dengan jelas | Berhasil |
| Pengacakan Soal Kuis | Tidak menemukan soal yang sama | Berhasil |
| Waktu Kuis | Waktu berjalan | Berhasil |

4. Kesimpulan dan Saran

Berdasarkan hasil dari penelitian dan pengujian algoritma *Fisher-Yates* telah menghasilkan permutasi pengacakan soal yang tidak bias sehingga tidak ada perulangan soal. Implementasi dari algoritma *Fisher-Yates* yang digunakan pada aplikasi penelitian ini berjalan dengan baik dan tidak terdapat perulangan soal kuis, aplikasi juga berhasil dioperasikan pada sistem operasi android yang berbeda-beda dan juga pengujian aplikasi dengan metode blackbox testing berhasil dilakukan dari awal hingga akhir aplikasi.

5. Daftar Pustaka

- [1] Hasan, M.A., Supriadi, S. and Zamzami, Z., 2017. Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru (Studi Kasus: Universitas Lancang Kuning Riau). *Jurnal Nasional Teknologi dan Sistem Informasi*, 3(2), pp.291-298.
- [2] Priantama, R. and Priandani, Y., 2019. Implementasi Algoritma Fisher Yates Untuk Pengacakan Soal Pada Aplikasi Mobile Learning Kuis Fiqih Berbasis Android. *Nuansa Informatika*, 13(2), p.40.
- [3] Asriyanik, A. and Apriyandari, W., 2020. Implementation of the Algorithm Fisher Yates Shuffle on Game Quiz Environment. *Journal Of Informatics and Telecommunication Engineering*, 4(1), pp.161-172.
- [4] Juniawan, F.P. and Hengki, H., 2019. Pengacakan Soal Ujian Penerimaan POLRI Menggunakan Algoritme Fisher Yates Shuffle. *Telematika*, 12(1), pp.1-13.
- [5] Maryono, I., Zulfikar, W.B. and Kariadinata, R., 2018. The implementation of fisher yates shuffle on aljabar learning media based on hybrid application. In *MATEC Web of Conferences* (Vol. 197, p. 01006). EDP Sciences.
- [6] Subaeki, B. and Ardiansyah, D., 2017. Implementasi Algoritma Fisher-Yates Shuffle Pada Aplikasi Multimedia Interaktif Untuk Pembelajaran Tenses Bahasa Inggris. *Infotronik: Jurnal Teknologi Informasi dan Elektronika*, 2(1).
- [7] Susilo, J. and Yenni, H., 2014. Perancangan Sistem Ujian Seleksi Penerimaan Mahasiswa Baru Dengan Menggunakan Algoritma Fisher-Yates Shuffle pada STMIK-AMIK RIAU. *Sains dan Teknologi Informasi*, 3(2), pp.27-32.
- [8] Marlindawati and D. N. Seputro., 2017. Penerapan Algoritma *Fisher-Yates* Shuffle Dengan Metode Modern Pada Try Out Ujian Semester, *Semin. Nas. Inov. Dan Teknol.*, pp. 2549–7952.
- [9] W. Gunawan and H. D. Wijaya., 2019. An Application of Multimedia for Basic Arabic Learning Using FisherYates Shuffle Algorithm on Android Based, *Sch. Bull.*, vol. 9771, pp. 347–355.
- [10] Ade-Ibijola, A.O., 2012. A simulated enhancement of Fisher-Yates algorithm for shuffling in virtual card games using domain-specific data structures. *International Journal of Computer Applications*, 54(11).
- [11] Tayel, M., Dawood, G. and Shawky, H., 2018, October. Block cipher S-box modification based on Fisher-Yates Shuffle and Ikeda Map. In *2018 IEEE 18th International Conference on Communication Technology (ICCT)* (pp. 59-64). IEEE.

- [12] S. A. Zuliani, E. Winarno, and Studi., 2018. Game Pembelajaran Membaca Iqra ' Menggunakan Metode Fisher Yates Shuffle, *Pros. SINTAK*, vol. 2, pp. 333–337.
- [13] Putra, R.R.C. and Sugihartono, T., 2019. Penerapan Algoritma Fisher-Yates Shuffle pada Computer Based Test Ujian Sekolah di SMKN 1 Payung. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 18(2), pp.276-283.
- [14] Haditama, I., Slamet, C. and Fauzy, D., 2016. Implementasi Algoritma Fisher-Yates Dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android. *Jurnal Online Informatika*, 1(1), pp.51-58.
- [15] Saeed, S., Umar, M.S., Ali, M.A. and Ahmad, M., 2014. Fisher-Yates chaotic shuffling based image encryption. *arXiv preprint arXiv:1410.7540*.
- [16] Hazra, T.K., Ghosh, R., Kumar, S., Dutta, S. and Chakraborty, A.K., 2015, October. File encryption using Fisher-Yates shuffle. In 2015 International Conference and Workshop on Computing and Communication (IEMCON) (pp. 1-7). IEEE.
- [17] Ramdania, D.R., Irfan, M., Habsah, S.N., Slamet, C., Uriawan, W. and Manaf, K., 2020. Fisher-Yates and fuzzy Sugeno in game for children with special needs. *Telkomnika*, 18(2).