

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

Kiki Vebiant^{a1}, Mohammad Iwan Wahyuddin^{a2}, Ratih Titi Komala Sari^{a3}

^a*Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional*

Jl. Sawo Manila, RT.14/RW.3, Ps. Minggu, Kec. Ps. Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta

¹kvebiant@gmail.com

²iwan.wahyuddi@civitas.unas.ac.id

³ratih.titi@civitas.unas.ac.id

Abstrak

Bahasa Inggris merupakan bahasa yang banyak digunakan pada berbagai negara-negara di dunia. Hal tersebut menjadikan bahasa Inggris menjadi internasional dan juga harus dipelajari pada sekolah maupun pada perguruan tinggi. Dalam 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.

Kata kunci: Android, *Fisher-Yates*, *Tenses*, Aplikasi Pembelajaran

Design of Learning Media for Android-Based English Tenses using Fisher-Yates Algorithm

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.

Keywords: Android, Fisher-Yates, Tenses, Learning Application

I. PENDAHULUAN

Bahasa Inggris merupakan bahasa Internasional yang banyak digunakan pada berbagai negara-negara di dunia. 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 bahasa 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].

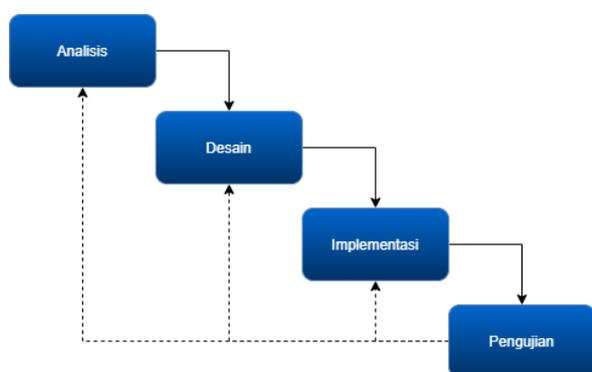
Adapula penelitian serupa dari Iyon Maryono dkk, dimana dalam penelitiannya memanfaatkan Algoritma Fisher-Yates untuk pengacakan pada kuis aljabar pada aplikasi hybrid yang menunjukkan bahwa algoritma Fisher-Yates sangatlah cocok untuk metode pengacakan kuis[5]. Kemudian juga terdapat penelitian serupa 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.

II. 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.

A. 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 I
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

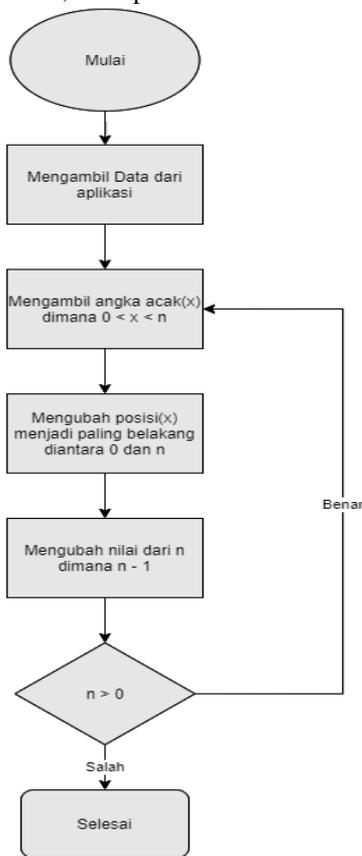
Tabel I. menjelaskan alur dari proses pengacakan algoritma Fisher-Yates yang diantaranya:

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.

B. 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.



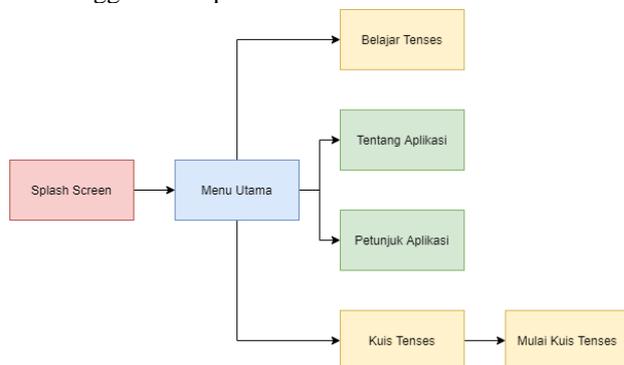
Gambar 2. Flowchart sistem

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.

C. 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.

III. HASIL DAN PEMBAHASAN

Pada tahapan ini merupakan tahap dimana analisis kebutuhan, pengimplementasi aplikasi hasil rancang desain ke pada aplikasi, juga pengujian sistem.

A. 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 III
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 II. 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 IIIII
KEBUTUHAN PERANGKAT LUNAK

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

Tabel III. 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.

B. Tampilan Aplikasi

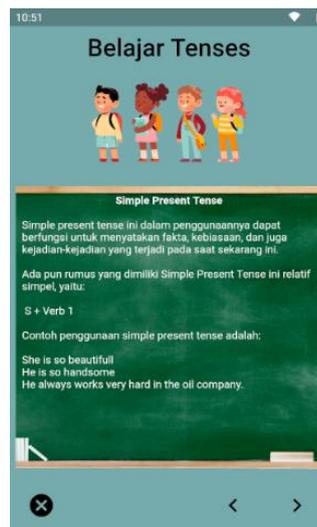
Dalam perancangan sistem aplikasi, peneliti menggunakan framework flutter untuk membuat desain, kode sistem aplikasi, logika dan lain sebagainya.



Gambar 4. Tampilan halaman menu utama

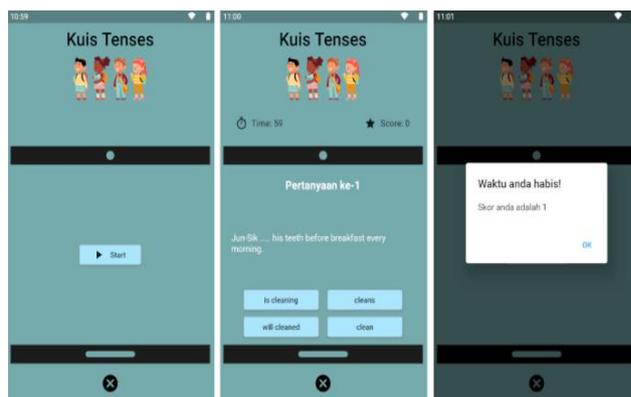
Gambar 4 merupakan tampilan halaman menu utama dari aplikasi yang terdiri dari 5 button yaitu button belajar tenses berfungsi untuk berpindah ke halaman belajar tenses, button quis tenses berfungsi untuk berpindah ke halaman kuis tenses, button tentang aplikasi yang berfungsi untuk memunculkan informasi tentang aplikasi, button petunjuk

yang berfungsi untuk memunculkan informasi petunjuk penggunaan aplikasi dan button silang untuk keluar dari aplikasi.



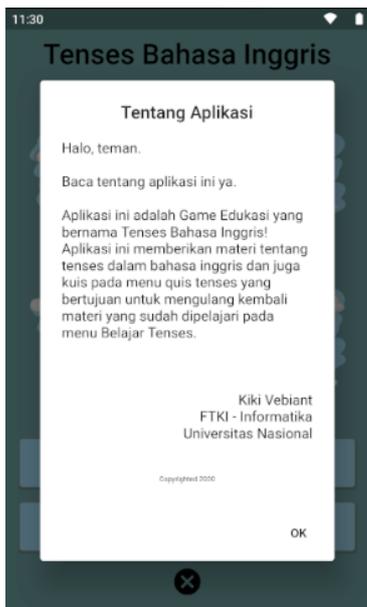
Gambar 5. Tampilan halaman belajar tenses

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 kembali, 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 kembali, selanjutnya setelah waktu telah habis akan menunjukkan informasi bahwa waktu telah habis dan berapa skor yang didapatkan lalu kembali 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.

C. Pengujian

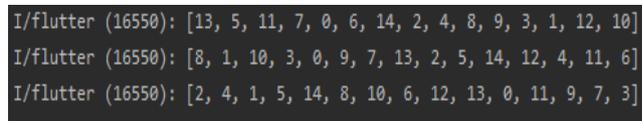
Pengujian pada penelitian ini dilakukan melalui beberapa tahap, diantaranya:

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 IV.

TABEL IVV
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. Pengujian algoritma Fisher-Yates pada aplikasi

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 V.

TABEL V
PENGUJIAN APLIKASI BLACKBOX TESTING

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

IV. KESIMPULAN

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.

DAFTAR PUSTAKA

- [1] M. A. Hasan, S. Supriadi, and Z. Zamzami, "Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru (Studi Kasus : Universitas Lancang Kuning Riau)," *J. Nas. Teknol. dan Sist. Inf.*, vol. 3, no. 2, pp. 291–298, 2017, doi: 10.25077/teknosi.v3i2.2017.291-298.
- [2] R. Priantama and Y. Priandani, "Implementasi Algoritma Fisher Yates Untuk Pengacakan Soal Pada Aplikasi Mobile Learning Kuis Fiqih Berbasis Android," *Nuansa Inform.*, vol. 13, no. 2, p. 40, 2019, doi: 10.25134/nuansa.v13i2.1951.
- [3] W. A. Rohmah, A. Asriyanik, and W. Apriyandari, "Implementation of the Algorithm Fisher Yates Shuffle on Game Quiz Environment," *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 161–172, 2020, doi: 10.31289/jite.v4i1.3863.
- [4] F. P. Juniawan *et al.*, "Pengacakan Soal Ujian Penerimaan Polri," *J. Telemat.*, vol. 1, no. 1, pp. 1–13, 2019.
- [5] I. Maryono, W. B. Zulfikar, and R. Kariadinata, "The implementation of fisher yates shuffle on aljabar learning media based on hybrid application," *MATEC Web Conf.*, vol. 197, pp. 1–5, 2018, doi: 10.1051/mateconf/201819701006.
- [6] B. Subaeki and D. Ardiansyah, "Implementasi Algoritma Fisher-Yates Shuffle Pada Aplikasi Multimedia Interaktif Untuk Pembelajaran Tenses Bahasa Inggris," *Infotronik J. Teknol. Inf. dan Elektron.*, vol. 2, no. 1, pp. 67–74, 2017, doi: 10.32897/infotronik.2017.2.1.31.
- [7] J. Susillo and H. Yenni, "Menggunakan Algoritma Fisher-Yates Shuffle pada STMIK-AMIK RIAU Joko Susilo," vol. 3, pp. 27–32, 2014.
- [8] Marlindawati and D. N. Seputro, "Penerapan Algoritma Fisher-Yates Shuffle Dengan Metode Modern Pada Try Out Ujian Semester," *Semin. Nas. Inov. Dan Teknol.*, pp. 2549–7952, 2017.
- [9] W. Gunawan and H. D. Wijaya, "An Application of Multimedia for Basic Arabic Learning Using FisherYates Shuffle Algorithm on Android Based," *Sch. Bull.*, vol. 9771, pp. 347–355, 2019, doi: 10.21276/sb.2019.5.7.6.
- [10] A. Olu, "A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures," *Int. J. Comput. Appl.*, vol. 54, no. 11, pp. 975–8887, 2012, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.114&rep=rep1&type=pdf>.
- [11] M. Tayel, G. Dawood, and H. Shawky, "Block cipher S-box modification based on fisher-yates shuffle and ikeda map," *Int. Conf. Commun. Technol. Proceedings, ICCT*, vol. 2019-October, pp. 59–64, 2019, doi: 10.1109/ICCT.2018.8600161.
- [12] S. A. Zuliani, E. Winarno, and Studi, "Game Pembelajaran Membaca Iqra ' Menggunakan Metode Fisher Yates Shuffle," *Pros. SINTAK*, vol. 2, pp. 333–337, 2018, [Online]. Available: <https://www.unisbank.ac.id/ojs/index.php/sintak/article/view/6639>.
- [13] R. R. C. Putra and T. Sugihartono, "Penerapan Algoritma Fisher-Yates Shuffle pada Computer Based Test Ujian Sekolah di SMKN 1 Payung," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 276–283, 2019, doi: 10.30812/matrik.v18i2.399.
- [14] I. Haditama, C. Slamet, and D. Fauzy, "Implementasi Algoritma Fisher-Yates Dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android," *J. Online Inform.*, vol. 1, no. 1, p. 51, 2016, doi: 10.15575/join.v1i1.11.
- [15] S. Saeed, M. S. Umar, M. A. Ali, and M. Ahmad, "Fisher-Yates Chaotic Shuffling Based Image Encryption," 2014, [Online]. Available: <http://arxiv.org/abs/1410.7540>.
- [16] T. K. Hazra, R. Ghosh, S. Kumar, S. Dutta, and A. K. Chakraborty, "File encryption using Fisher-Yates Shuffle," *2015 Int. Conf. Work. Comput. Commun. IEMCON 2015*, no. March 2016, 2015, doi: 10.1109/IEMCON.2015.7344521.
- [17] D. R. Ramdania, M. Irfan, S. N. Habsah, C. Slamet, W. Uriawan, and K. Manaf, "Fisher-Yates and fuzzy Sugeno in game for children with special needs," *Telkonnika (Telecommunication Comput. Electron. Control)*, vol. 18, no. 2, pp. 879–889, 2020, doi: 10.12928/TELKOMNIKA.V18I2.14906.