

Implementasi Algoritme *Fisher Yates Shuffle* untuk Pengacakan Soal Tes IQ Berbasis Android

Muhammad Tofa Nurcholis^{1*}, Arif Dwi Laksito², Hafizh Fauzi Fathurahman³
 Informatika, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
 *e-mail Corresponding Author. tofa@amikom.ac.id

Abstract

This study aims to implement the Fisher Yates Shuffle algorithm on an Android-based IQ test application to ensure random and unbiased randomization of questions. Fisher Yates Shuffle Algorithm is one of the randomization algorithms used to reduce bias and ensure good random distribution in randomizing elements in an array. The results of this study indicate that the implementation of the Fisher Yates Shuffle algorithm in an Android-based IQ test application ensures that the questions displayed to the user are always in random order, so that there is no bias in the test results. Ease of use also improves with this implementation.

Keywords: *Fisher Yates Shuffle Algorithm; Implementations; IQ test.*

Abstrak

Penelitian ini bertujuan untuk mengimplementasikan algoritme *Fisher Yates Shuffle* pada aplikasi tes IQ berbasis Android untuk memastikan pengacakan soal yang acak dan tidak bias. Algoritme *Fisher Yates Shuffle* merupakan salah satu algoritme pengacakan yang digunakan untuk mengurangi bias dan memastikan distribusi acak yang baik dalam pengacakan elemen dalam suatu *array*. Hasil dari penelitian ini menunjukkan bahwa implementasi algoritme *Fisher Yates Shuffle* pada aplikasi tes IQ berbasis Android memastikan bahwa soal yang ditampilkan kepada pengguna selalu dalam urutan acak, sehingga tidak ada bias dalam hasil tes. Kemudahan penggunaan juga meningkat dengan implementasi ini.

Kata kunci: *Algoritme Fisher Yates Shuffle; Implementasi; Tes IQ*

1. Pendahuluan

Wabah Virus Covid-19 pertama kali muncul atau di temukan di kota Wuhan. Sudah tercatat lebih dari 200 negara yang ada di dunia melaporkan adanya kenaikan kasus Covid-19, adanya virus Covid-19 pada tahun 2020 memberi dampak yang luar biasa hampir pada aspek semua bidang, salah satunya adalah bidang Pendidikan [1].

Pendidikan merupakan suatu usaha yang dilakukan untuk mengembangkan kemampuan dan membentuk watak serta peradaban bangsa yang bermartabat dalam rangka mencerdaskan kehidupan bangsa. Berdasarkan Permen RI No. 47 pasal 2 tahun 2008 tentang wajib belajar, wajib belajar berfungsi mengupayakan perluasan dan pemerataan kesempatan memperoleh Pendidikan bermutu bagi setiap warga Indonesia. Hasil belajar pada hakekatnya merupakan pencerminan dari usaha belajar. Hasil belajar secara umum dapat dipengaruhi beberapa hal, salah satunya adalah faktor psikologi, secara psikologi perilaku berkarakter merupakan perwujudan dari potensi *intelligence Quotient (IQ)* [2].

Intelligence Quotient (IQ) adalah alat ukur kemampuan intelektual, Analisa, logika dan raiso seseorang. *IQ* mengukur kecepatan untuk mempelajari hal-hal baru, memusatkan perhatian pada aneka tugas dan latihan, menyimpan dan mengingat Kembali informasi secara objektif. Untuk mengetahui seberapa besar *IQ* dapat dilakukan dengan cara uji *Tes IQ* [2].

Tes IQ pada umumnya masih bersifat tertulis atau manual sehingga untuk mendapatkan hasil *tes IQ* membutuhkan waktu lama serta biaya yang mahal, tetapi ini sudah terpecahkan dengan adanya aplikasi *Tes IQ* di Android yang bisa di unduh di Play Store dan *App Store*. Namun kenyataannya kebanyakan aplikasi *Tes IQ* masih menggunakan soal yang sama, sehingga user dapat memanipulasi hasil dari *Tes IQ*, sehingga hal ini tidak dapat memecahkan masalah kecerdasan pada seseorang.

Untuk memecahkan permasalahan diatas maka dibuatkan sistem pengacakan dengan mengimplementasikan sebuah *algoritma* yaitu algoritme *Fisher-yates shuffle*. Salah satu contoh penelitian yang membuktikan bahwa algoritme *Fisher-Yates Shuffle* lebih baik dari algoritme pengacakan lain adalah "Implementation and Performance Evaluation of Random Number Generation Algorithms for Mobile Devices" oleh Y. Chen and X. Liu [3]. Dalam penelitian ini, Fisher-Yates Shuffle dibandingkan dengan beberapa algoritme pengacakan lain pada perangkat mobile dan hasilnya menunjukkan bahwa Fisher-Yates Shuffle memiliki tingkat akurasi yang lebih baik dalam mengacak urutan elemen.

Tujuan penelitian ini untuk mengimplementasikan *algoritme fisher yates shuffle* untuk mengacak sebuah soal tes IQ berbasis Android yang menyediakan layanan tes IQ dengan menampilkan urutan soal secara acak, sehingga user tidak dapat memanipulasi dalam melukan tes IQ serta mendapatkan hasil murni dalam melakukan Tes IQ.

2. Tinjauan Pustaka

Implementasi Algoritme *Fisher Yates Shuffle* untuk Pengacakan Soal Tes IQ Berbasis Android adalah penelitian yang membahas tentang penerapan algoritme *Fisher Yates Shuffle* pada aplikasi pengacakan soal tes IQ berbasis Android. Algoritme *Fisher Yates Shuffle* merupakan metode pengacakan yang efisien dan memiliki tingkat acak yang baik [4].

Beberapa studi sebelumnya membuktikan bahwa penerapan algoritme *Fisher Yates Shuffle* dapat meningkatkan tingkat validitas dan reliabilitas hasil tes IQ. Seperti yang dibahas oleh Nguyen et al. [5], dalam penelitian mereka tentang penerapan algoritme *Fisher Yates Shuffle* pada aplikasi pengacakan soal tes konsep matematika, mereka menemukan bahwa implementasi algoritme *Fisher Yates Shuffle* menghasilkan tingkat acak yang baik dan memiliki tingkat validitas dan reliabilitas yang lebih tinggi dibandingkan dengan metode pengacakan manual [5].

Penelitian terdahulu yang membahas tentang pengacakan soal tes IQ umumnya menggunakan metode pengacakan manual atau menggunakan aplikasi pengacakan yang tidak menggunakan algoritme *Fisher Yates Shuffle*. Beberapa studi yang melakukan implementasi aplikasi pengacakan soal tes IQ tanpa menggunakan algoritme Fisher Yates Shuffle antara lain adalah penelitian oleh Wu et al. [6] yang menggunakan aplikasi pengacakan soal tes IQ berbasis web. Hasil dari penelitian tersebut menunjukkan bahwa aplikasi pengacakan soal tes IQ berbasis web tersebut tidak efisien dan tidak memiliki tingkat acak yang baik.

Penelitian Febrian dkk [7] yang berjudul Algoritme Linear Congruent Method dan Algoritme *Fisher Yates Shuffle* Pada Kuis Ketangkasan Berbasis Android membahas tentang implementasi dua algoritme tersebut dalam sebuah aplikasi kuis ketangkasan berbasis Android. *Linear Congruent Method* digunakan untuk menghasilkan bilangan acak, sedangkan *Fisher Yates Shuffle* digunakan untuk melakukan pengacakan pada elemen-elemen dalam suatu data. Implementasi kedua algoritme ini dapat membantu membuat aplikasi kuis ketangkasan yang lebih interaktif dan menyenangkan bagi pengguna.

Penelitian Rohmah [8] membahas implementasi algoritme *Fisher-Yates Shuffle* dalam lingkungan permainan kuis. Algoritme ini digunakan untuk mengacak urutan pertanyaan dalam permainan kuis untuk memastikan keadilan dan variabilitas dalam permainan. Artikel ini melaporkan hasil implementasi algoritme *Fisher-Yates Shuffle* dan bagaimana hal tersebut mempengaruhi pengalaman bermain kuis.

Dalam paper Zhang and Chen [9] "A Study of Array Shuffling Algorithms for Mobile Devices," pembahasan utama adalah analisis dan studi tentang berbagai algoritme pengacakan array untuk perangkat mobile. Jurnal ini menjelaskan berbagai metode pengacakan array dan membandingkan kinerja mereka pada perangkat *mobile*.

Dalam paper Chen and Liu [3], "Implementation and performance evaluation of random number generation algorithms for mobile devices," pembahasan utama adalah implementasi dan evaluasi kinerja berbagai algoritme pembuatan nomor acak untuk perangkat mobile. Jurnal ini menjelaskan bagaimana *algoritme-algoritme* tersebut diimplementasikan pada perangkat mobile dan bagaimana hasil kinerjanya dibandingkan satu sama lain.

Dalam Paper Kwon and Lee [10], "Implementation of the Fisher-Yates Shuffle Algorithm in Android," pembahasan utama adalah implementasi algoritme *Fisher-Yates Shuffle* pada sistem operasi Android. Jurnal ini menjelaskan bagaimana algoritme ini diimplementasikan pada perangkat Android dan menunjukkan bagaimana kinerjanya dibandingkan dengan algoritme pengacakan lain.

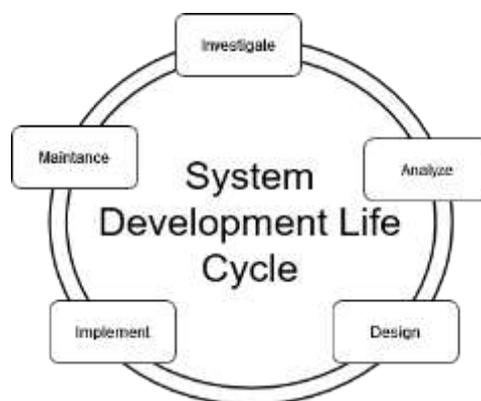
Dalam Paper Kim and Park [11], "An efficient random number generator for mobile devices based on Fisher-Yates shuffle algorithm," pembahasan utama adalah pembuatan generator nomor acak efisien untuk perangkat mobile berdasarkan algoritme Fisher-Yates Shuffle. Jurnal ini menjelaskan bagaimana algoritme Fisher-Yates Shuffle digunakan untuk membuat generator nomor acak efisien untuk perangkat mobile dan bagaimana kinerjanya dibandingkan dengan generator nomor acak lain.

Berdasarkan beberapa penelitian sebelumnya *Algoritme Fisher Yates Shuffle* adalah salah satu algoritme populer yang digunakan untuk mengacak urutan elemen dalam suatu array. Ini memastikan distribusi acak yang baik dan efisiensi yang baik dalam hal waktu eksekusi. Dalam era digital saat ini, aplikasi pada platform Android menjadi sangat populer dan banyak digunakan. Oleh karena itu, membangun aplikasi untuk mengacak soal tes IQ berbasis Android merupakan solusi yang inovatif dan praktis. Dalam penelitian *Algoritme Fisher Yates Shuffle* digunakan untuk mengacak soal Test IQ. Dengan mengacak soal tes IQ, tes tersebut dapat lebih adil dan objektif karena setiap individu akan mendapatkan soal yang berbeda. Kombinasi Algoritme Fisher Yates Shuffle dan Aplikasi Android: Implementasi algoritme Fisher Yates Shuffle pada aplikasi Android akan membuat proses pengacakan soal tes IQ menjadi lebih efisien dan mudah dilakukan. Aplikasi ini juga memiliki kelebihan seperti portabilitas dan aksesibilitas yang lebih baik dibandingkan dengan metode tradisional. Berdasarkan state of the art di atas, implementasi algoritme *Fisher Yates Shuffle* pada aplikasi pengacakan soal tes IQ berbasis Android merupakan solusi yang inovatif, efisien, dan praktis untuk meningkatkan objektivitas dan efektivitas tes IQ.

3. Metodologi

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam perancangan aplikasi android serta pengimplementasian algoritme *Fisher Yate Shuffle* ini dilakukan berdasarkan SDLC (*System Development Life Cycle*), dengan menggunakan SDLC dapat berjalan sesuai dengan yang diharapkan. Tahapan-tahapan SDLC yaitu *Planning* atau *Investigate*, *Analyze*, *Design*, *Implement*, *Testing / Maintance*.



Gambar 1. Tahapan dalam SDLC

SDLC adalah sebuah metode yang digunakan untuk mengembangkan sebuah system. SDLC merupakan sebuah proses logika yang digunakan oleh seseorang system analyst untuk mengembangkan sebuah system informasi yang melibatkan requirements, validation, serta training.

SDLC identik dengan teknik pengembangan sistem *Waterfall*, karena tahapannya menurun dari atas kebawah. Ada beberapa tahapan dalam SDLC yaitu [12]:

1) Tahap Planning (Perencanaan)

Dalam perencanaan sebuah website diperlukan sebuah perencanaan yang matang yaitu dengan melakukan studi kelayakan tentang metode yang digunakan pada proses pengumpulan data. Studi kelayakan yang dilakukan yaitu metode pengumpulan data, sumber data, dan teknik Analisa data.

2) Tahap Analisis

Pada tahapan ini dilakukan identifikasi dan analisis kebutuhan aplikasi terkait pengacakan soal tes IQ untuk setiap user yang berbeda. Untuk sumber data berupa soal, yang pertama diambil dari modul Tes Intelegensi yang disusun oleh Nur Habibah sebagai pedoman sebelum pembuatan soal Tes IQ. Sumber kedua sebagai acuan soal Tes IQ yaitu Tes deret angka, deret huruf Tes Sinonim, Tes Antonim dan Tes analogi verbal dengan mengambil 20 data soal permasing-masing tes, sumber ini diambil dari buku yang berjudul Best Book Psikotes yang disusun oleh Aziz, Prasetya, dan Dedi [13, 14].

3) Tahap Perancangan

1. Desain *Database*: Desain dan implementasi database yang sesuai dengan spesifikasi yang ditentukan.
2. Desain *User Interface*: Desain tampilan dan navigasi aplikasi agar mudah digunakan oleh pengguna.

4) Tahap Implementasi

Implementasi aplikasi Tes IQ berbasis android dilakukan sesuai dengan perancangan oleh sistem. Implementasi dilakukan dengan membuat sebuah aplikasi, database, penulisan source code, serta penerapan sebuah algoritme fisher yates shuffle dalam sebuah aplikasi yang telah dibuat sehingga aplikasi dapat digunakan sesuai dengan tujuan utama dalam pembuatan aplikasi dan pengimplementasian algoritme didalam sebuah aplikasi tersebut.

5) Tahap Pengujian

Pengujian sistem pada aplikasi pengacakan soal tes iq berbasis android dilakukan dengan maksud untuk menemukan kesalahan pada suatu sistem. Pengujian dalam sistem ini ada 2 macam pengujian yaitu:

a. Pengujian Validitas

Pengujian Validitas merupakan sebuah pengujian yang diarahkan untuk menentukan kekuatan sistem dan dilakukan dengan mengambil sebuah sample dari responden untuk menilai sebuah sistem apakah sistem tersebut sesuai dengan hasil yang dibutuhkan atau tidak.

b. Pengujian Fungsional

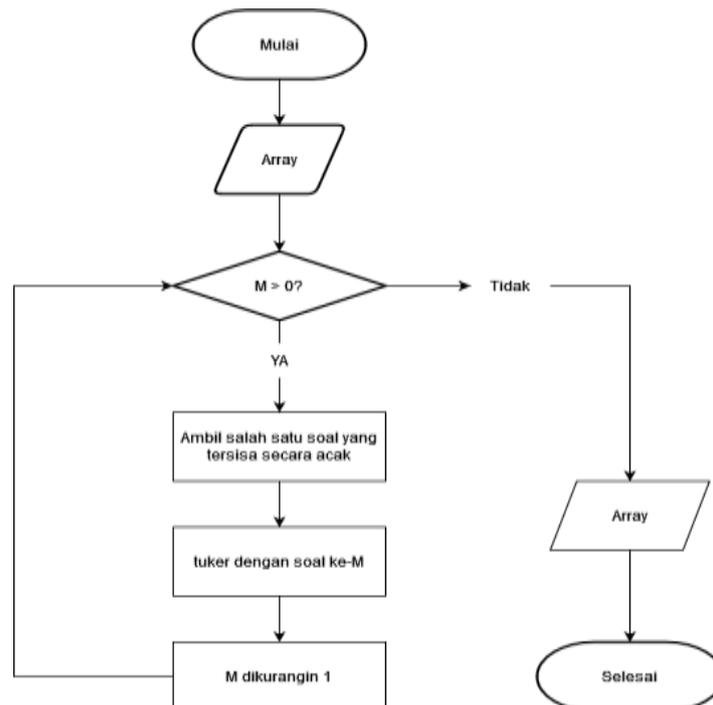
Merupakan pengujian pada proses input dan output. Pengujian program menggunakan pengujian secara fungsional yang mana digunakan untuk menguji fungsi dari sebuah menu serta pengimplementasian sebuah algoritme yang terdapat didalam sebuah aplikasi pengacakan soal Tes IQ dengan mengimplementasikan algoritme fisher yates shuffle berbasis android. Pada pengetesan ini kebenaran sistem yang diuji dilihat berdasarkan keluaran atau output yang dihasilkan. Seperti pada proses penginputan login maka akan dihasilkan sebuah data sesuai dengan masukan data saat login, jika data tersebut salah berarti data yang dimasukkan saat login tidak sesuai. Yang kedua pada proses pengimplementasian algoritme fisher yates shuffle, jika source code yang dituliskan saat pembuatan sebuah sistem maka output yang dihasilkan adalah sebuah pengacakan soal yang sesuai, jika dalam sebuah soal didalam sistem masih belum teracak maka source code yang diberikan masih belum berfungsi.

3.2 Algoritme Fisher Yates Shuffle

Teori Algoritme *Fisher-Yates Shuffle* adalah metode acak (*shuffling*) data yang ditemukan oleh Richard Fisher dan Ronald Yates pada tahun 1938. Algoritme ini menjamin distribusi yang acak dan uniform dari data yang diacak. Ini digunakan untuk mencampur data seperti urutan kartu dalam permainan kartu, atau mengurutkan angka acak dalam aplikasi lain. Berikut adalah algoritme Fisher-Yates Shuffle [15, 16]:

- 1 Mulai dari indeks terakhir dalam urutan data
- 2 Pilih indeks acak antara indeks saat ini dan indeks pertama
- 3 Tukar elemen pada indeks saat ini dengan elemen pada indeks acak yang dipilih
- 4 Ulangi langkah 2 dan 3 sampai indeks saat ini mencapai indeks pertama

Proses pengimplementasian pada algoritme *Fisher Yates Shuffle* pada sebuah sistem aplikasi mobile untuk pengacakan soal IQ, seperti pada Gambar 2.



Gambar 2. Flowchart Algoritme Fisher Yates Shuffle

Prosedur pada Gambar 2 disajikan sebagai berikut:

1. Menghitung jumlah soal yang ada.
2. Jika jumlah soal lebih dari 0, maka,
3. Mengambil satu soal yang tersisa secara acak
4. Menukar dengan soal ke-m (m yaitu batasan jumlah soal, semisal ada 30 isi soal berarti tukar ke isi soal nomor 30).
5. Lalu jumlah soal dikurangi 1.
6. Perulangan akan terus berlanjut sampai isi soal yang belum teracak sudah habis.
7. Jika jumlah isi soal dibawah 0 maka tidak akan melakukan pengacakan.
8. Selesai.

3.3 Bahan Penelitian

Data yang didapat yaitu data soal yang berbeda dalam pengacakannya, sehingga user tidak akan menerima soal dengan urutan yang sama

4. Hasil dan Pembahasan

4.1 Analisis Kebutuhan

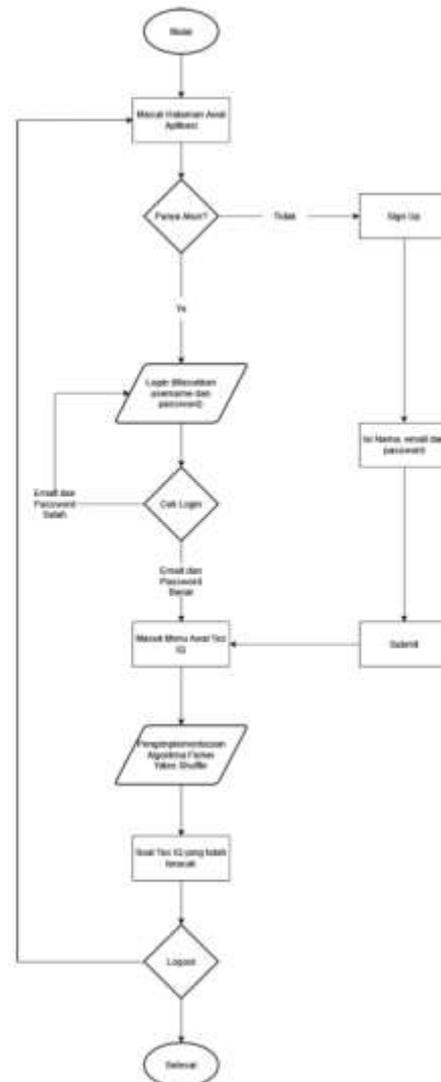
Dalam proses menjalankan aplikasi tes iq, sistem mengambil data orang yang sudah pernah login dan baru mendaftarkan akun yang terdapat didalam database SQLite. Kemudian system harus mampu melakukan pengacakan soal test untuk setiap user yang berbeda.

4.2 Perancangan

a) Model Database

Perancangan database meliputi pembentukan struktur tabel yang akan digunakan oleh system. Didalam perancangan sebuah database, disini hanya menggunakan tabel user dalam pembuatan database, tabel user berisi id_user, username dan password. Tabel tersebut digunakan untuk menyimpan suatu data username dan password dari user.

b) Model Logik Sistem



Gambar 3. Flowchart Pengoperasian Sistem Aplikasi

1. User masuk pada halaman awal aplikasi tes IQ
2. Jika mempunyai akun, maka langsung login, jika tidak maka melakukan sign up untuk mempunyai akun
3. Sistem akan mengecek login terlebih dahulu, jika benar maka langsung bisa login ke halaman utama tes IQ, jika tidak maka sistem tidak bisa lanjut untuk login
4. Pada saat masuk ke halaman utama tes IQ, maka user bisa langsung memulai tes, dan saat itu juga sistem mengimplementasikan algoritme Fisher Yates Shuffle.
5. Setelah melakukan pengimplementasian, maka soal tes iq sudah teracak
6. Saat user logout sistem akan membawanya Kembali ke menu login

4.3 Implementasi Sistem

Perancangan sistem merupakan tahapan setelah analisis sistem yang didefinisikan dari kebutuhan fungsional dan persiapan perancangan yang menggambarkan bagaimana sistem di bentuk, serta bagaimana penerapana sebuah algoritme didalam aplikasi tersebut, yang berupa perancangan, penggambaran, pengimplementasian, dan elemen-elemen yang terpisah kedalam suatu kesatuan yang utuh dan menyangkut konfigurasi dari beberapa komponen

perangkat keras dan perangkat lunak. Aplikasi yang dibuat ini akan digunakan penulis dengan membuat akun, sehingga seseorang dapat mengakses aplikasi tersebut untuk mengerjakan soal Tes IQ. Bahasa yang digunakan adalah Bahasa Java. Disini penulis hanya berfokus pada pengimplementasian algoritma, pengimplementasian sistem hanya dijelaskan secara umum saja.

Berdasarkan hasil analisis dan perancangan, maka diimplementasikan sebuah aplikasi pengacakan soal tes Iq berbasis android, sistem yang penulis rancang dan buat merupakan salah satu solusi yang ditawarkan untuk melakukan uji coba tes iq dengan soal yang teracak. Dengan merancang sistem aplikasi pengacakan soal tes iq berbasis android ini diharapkan berguna dan membantu dalam menguji tes iq user yang dimana soal yang ditampilkan tidak sama atau teracak sehingga user dapat mendapatkan hasil yang sesuai dengan kapasitas yang dimiliki user.

4.4 Implementasi algoritme *Fisher Yates Shuffle* pada soal Tes IQ

Pengimplementasian Algoritme didalam soal tes IQ merupakan tahap inti dimana proses terjadinya pengacakan ada pada tahap ini.

```
pertanyaanLists = PertanyaanBank.getPertanyaan(getSelectedTopicName);
Collections.shuffle(pertanyaanLists);
```

Gambar 4. Pengimplementasian Algoritme *Fisher Yates Shuffle*

Kemudian proses setelah proses pengimplementasian maka soal tes IQ secara otomatis akan teracak.



Gambar 5. Tampilan Setelah Pengacakan

Gambar 5 menunjukkan bahwa tampilan tersebut sesudah menggunakan pengimplementasian dari sebuah algoritme *Fisher Yates Shuffle*, ini bisa dilihat dari nomer angka soal dan nomer soal yang berbeda. Penggunaan algoritme ini sangat penting melihat tampilan soal teracak secara random.

4.5 Pengujian Sistem

Rencana pengujian merupakan proses menjalankan dan mengevaluasi perangkat lunak secara otomatis maupun manual guna menguji apakah perangkat lunak tersebut telah memenuhi persyaratan atau belum. Tujuan pengujian ini yaitu menilai apakah perangkat lunak yang dirancang sudah memenuhi kebutuhan pengguna.

Pengujian dari sebuah aplikasi pengacakan soal IQ dilakukan melalui seseorang sebagai user. Pada pengujian perangkat lunak menggunakan metode pengujian berupa *Black box*, karena pengujian *black box* mempunyai keuntungan yaitu:

- Digunakan untuk menguji fungsi khusus dari perangkat lunak yang dirancang.
- Kebenaran perangkat lunak diuji berdasarkan keluaran yang dihasilkan.
- Digunakan untuk menguji fungsi implementasi dari sebuah algoritme pada perangkat lunak.
- Kebenaran dari algoritme diuji berdasarkan keluaran yang dihasilkan,
- Kemampuan program dapat diukur dan diketahui dari kesalahan- kesalahan berdasarkan keluaran yang dihasilkan.

Dengan adanya pengujian *Black box* ini diharapkan jika ada kekurangan maupun kesalahan didalam aplikasi maupun pengimplementasian dari sebuah algoritme dapat diketahui sedini mungkin oleh peneliti. Pengujian sistem informasi dan implementasi algoritme dalam sebuah aplikasi pengacakan tes IQ berbasis android dengan berupa data masukan user pada aplikasi yang dibuat.

Tabel 1. Pengujian *Black Box*

Item Pengujian	Sasaran	Deskripsi
Proses Awal tanpa Pengimplementasian	Tanpa Algoritme	Memeriksa apakah soal terurut tanpa menggunakan pengimplemntasian ataukah sudah teracak
Proses pengimplementasian pada soal	Menggunakan algoritme	Memeriksa apakah soal teracak dengan menggunakan pengimplementasian algoritme
	User 1	Memeriksa apakah soal teracak dan didapatkan user sama dengan user lainnya, memeriksa urutan soal berapa saja yang keluar
	User 2	Memeriksa apakah soal teracak dan didapatkan user sama dengan user lainnya, memeriksa urutan soal berapa saja yang keluar
	User 3	Memeriksa apakah soal teracak dan didapatkan user sama dengan user lainnya, memeriksa urutan soal berapa saja yang keluar

Tabel 2. Tampilan Soal Tanpa Pengimplementasian Algoritme

Diskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang didapat	Keterangan
Proses Awal tanpa Pengimplementasian	Menjawab dan mengeklik soal selanjutnya	Soal akan urut dari nomor 1 sampai no terakhir	Berhasil	Nomor soal urut dari nomor 1, 2, 3, 4, 5-20

Tabel 3. Tampilan Soal Menggunakan Pengimplementasian Algoritme

Diskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang didapat	Keterangan
Proses Pengimplementasian Soal	Menjawab dan mengeklik soal selanjutnya	Soal akan teracak dari nomor 1 sampai no terakhir	Berhasil	Urutan soal acak no 20,2,15,12,18, 3,6,9,19,7, 5,14,1,16,11, 17,4,8,10,13

Tabel 4. Pengimplementasian Soal Terhadap User

Diskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang didapat	Keterangan
Proses Pengimplementasian Soal terhadap user 1	Mengeklik List Soal kemudian masuk ke soal kemudian klik back dan masuk lagi	Soal akan berbeda setiap selesai mengerjakan atau mengeklik kembali	Berhasil	Urutan soal saat sehabis mulai start tes adalah no 14,20,2,1,10, 3,16,17,15,9

Diskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang didapat	Keterangan
				Urutan soal setelah mengeklik Kembali atau mulai tes lagi : 19,16,8,17,18, 10,20,14,6,4
Proses Pengimplementasian Soal terhadap user 2	Mengeklik List Soal kemudian masuk ke soal kemudian klik back dan masuk lagi	Soal akan berbeda setiap selesai mengerjakan atau mengeklik kembali	Berhasil	Urutan soal saat sehabis mulai start tes adalah no 8,10,6,5,13, 20,1,3,17,15 Urutan soal setelah mengeklik Kembali atau mulai tes lagi : 18,13,17,5,4, 10,7,11,8,20
Proses Pengimplementasian Soal terhadap user 3	Mengeklik List Soal kemudian masuk ke soal kemudian klik back dan masuk lagi	Soal akan berbeda setiap selesai mengerjakan atau mengeklik kembali	Berhasil	Urutan soal saat sehabis mulai start tes adalah no 16,15,14,18,17, 4,9,3,8,7 Urutan soal setelah mengeklik Kembali atau mulai tes lagi : 4,12,2,14,16, 1,13,5,9,11

5. Simpulan

Pengimplementasian aplikasi pengacakan Tes IQ melalui lima tahap yaitu: Perancangan, Analisa, Desain Implementasi, Pengujian. Menghasilkan sebuah pengacakan soal tes IQ dengan menggunakan pengimplementasian sebuah algoritme fisher yates shuffle dalam sebuah aplikasi android. Dalam pengimplementasian algoritme *Fisher Yates Shuffle* pada aplikasi tes IQ menghasilkan soal yang berbeda dalam pengacakannya, sehingga user tidak akan menerima soal dengan urutan yang sama dan implementasi ini telah dirancang sesuai dan dapat digunakan dengan baik. Hasil dari penelitian ini memperkuat keyakinan bahwa algoritme *Fisher Yates Shuffle* efektif digunakan untuk pengacakan soal tes IQ pada perangkat berbasis Android. Hasil dari penelitian ini juga bisa membantu memperkuat teori atau model yang ada dalam bidang terkait, dan membuka peluang untuk penelitian lanjutan yang memperluas atau memperbaiki temuan sebelumnya.

Daftar Referensi

- [1] R. Yunitasari and U. Hanifah, "Pengaruh Pembelajaran Daring terhadap Minat Belajar Siswa pada Masa COVID-19," *Jurnal Ilmu Pendidikan*, vol. 2, no. 1, pp. 232-243, 2020.
- [2] P. Iswahyudi, S. D. Sulung, D. Haricahyo and H. E. Martono, "Pengaruh Intelegence Quotient (IQ), Emotional Quotient (EQ) dan Spiritual Quotients (SQ) Terhadap Prestasi Belajar Mata Kuliah Pendidikan Pancasila pada Taruna Diplomat III penerbangan Sayap Tetap Angkatan I Di Akademik Penerbangan Indonesia Banyuwangi," *Jurnal Aviasi Indonesia*, vol. 1, no. 1, pp. 7-13, 2021

- [3] Y. Chen and X. Liu, "Implementation and performance evaluation of random number generation algorithms for mobile devices," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 4135-4147, 2019
- [4] D.E. Knuth, *The art of computer programming*. Addison-Wesley Longman Publishing Co., Inc., 1997
- [5] T.T. Nguyen, & T.A. Nguyen, "Implementation of Fisher-Yates Shuffle Algorithm for Randomizing Mathematical Concepts Test Questions". *Journal of Education and Practice*, vil. 11, no. 3, pp. 82-88, 2020.
- [6] X. Wu, Y. Jiang, & Y. Liu, "Research on the Random Question Generation System for IQ Tests Based on Web". *International Journal of Emerging Technologies in Learning (IJET)*, vol. 14, no. 12, pp. 85-92, 2019
- [7] R. Pebrian, Fauziah and I. D. Sholihati, "Algoritme Linear Congruent Method dan Algoritme Fisher Yates Shuffle Pada Kuis Ketangkasan Berbasis Android," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. 6, no. 02, pp. 167-177, 2021.
- [8] W. A. Rohmah, Asriyanik and W. Apriyandari, "Implementation of the Algorithm Fisher Yates Shuffle on Game Quiz Environment," *Journal of Informatics and Telecommunication Engineering*, vol. 1, no. 2, pp. 161-172, 2020.
- [9] X. Zhang and Q. Chen, "A Study of Array Shuffling Algorithms for Mobile Devices," *Journal of Mobile Information Systems*, pp. 1-13, 2021.
- [10] O. Kwon and J. Lee, "Implementation of the Fisher-Yates Shuffle Algorithm in Android," *International Journal of Mobile Computing and Applications*, vol. 8, no. 3, pp. 1-9, 2017.
- [11] S. Kim and J. Park, "An efficient random number generator for mobile devices based on Fisher-Yates shuffle algorithm," *Journal of Mobile Information Systems*, vol. 11, no. 1, pp. 1-12, 2015.
- [12] B. Soeherman and M. Pinontoan, *Designing Information System*, Jakarta: PT. Elex Media Komputindo, 2008.
- [13] N. Habibah, *Modul Tes Intelegensi*, Sidoarjo: UMSIDA, 2018
- [14] T. P. Salemba, A. A. Saefudin, P. A. Nugroho and D. Gunarto, *Best Book Psikotes Terlengkap*, Jakarta Selatan: PT. Bintang Wahyu, 2019.
- [15] R. A. Fisher and F. Yates, "Statistical tables for biological, agricultural and medical research," London: Longmans, Green and co, 1938.
- [16] Durstenfeld, "Algorithm 235: Random permutation," *Communications of the ACM*, vol. 7, no. 7, pp. 420-422, 1964.