

Algoritma Fisher-Yates Shuffle Pada Game Edukasi Jumble Hijaiyah

Albert Ramadhan¹⁾, Ratih Titi Komala Sari²⁾

¹⁾Teknik Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional

Correspondence author: Albert Ramadhan, albertramadhan321@gmail.com

DOI :<https://doi.org/10.37012/jtik.v8i1.759>

Abstrak

Seiring perkembangan zaman, banyak media pembelajaran tergantikan menjadi teknologi digital. Begitu pula dengan media pembelajaran yang bisa digunakan berupa *game* edukasi untuk melatih daya ingat. Permainan *jumble* bertujuan untuk memasang dua gambar yang sama pada seluruh kartu yang ada di *board*. *Game* bertujuan untuk meningkatkan kemampuan dalam melatih daya ingat dan meningkatkan konsentrasi pada saat memainkan game. Untuk metode menggunakan *Game Development Life Cycle* (GDLC) dalam melakukan pembuatan aplikasi dari awal hingga akhir dengan alur yang terstruktur. Pemograman pada *game* ini juga mengimplementasikan Algoritma *Fisher Yates Shuffle* untuk melakukan pengacakan objek yang ada pada *board*. Penggunaan algoritma ini memudahkan program mengacak jumlah kartu yang ada pada *board* untuk dijadikan tantangan bagi pemain menebak dimana gambar yang sama pada kartu tersebut. Permainan ini juga tersedia bermacam *level*, yaitu *Easy*, *Normal*, *Hard*, dan *Very Hard*. Dengan adanya tingkatan kesusahan pada game ini kita bisa melihat sejauh mana kita bisa mengingat objek dan seberapa cepat pemain bisa menyelesaikannya.

Kata Kunci: *Jumble* ; Permainan ; Algoritma *Fisher Yates Shuffle* ; *Game Development Life Cycle* ; *Level*

Abstract

Along with the times, many learning media have been replaced by digital technology. Likewise with learning media that can be used in the form of educational games to train memory. Mixed game aims to match two of the same image on all the cards on the board. The game aims to improve the ability to train memory and improve concentration when playing games. For the method of using Game Development Life Cycle (GDLC) in making applications from start to finish with a structured flow. Programming in this game also implements the Fisher Yates Shuffle Algorithm to randomize objects on the board. This use makes it easier for the program to randomize the number of cards on the board to be used as a challenge for players to guess where the same image is on the card. This game is also available in various levels, namely Easy, Normal, Hard, and Very Hard. With the levels in this game we can see how far we can remember objects and how fast players can complete them.

Keywords: *Jumble* ; *Game* ; *Algorithm Fisher-Yates Shuffle* ; *Game Development Life Cycle* ; *Level*

PENDAHULUAN

Menurut (Wijaya & Apridiansyah, 2020) mengatakan bahwa kemajuan teknologi pada industri dan bisnis mempengaruhi pengembangan *video game* juga sudah berkembang pesat. Banyak sekali yang bisa didapatkan dari game seperti sebagai sarana hiburan maupun pendidikan bagi anak muda maupun kalangan dewasa, pria maupun wanita, dan juga untuk anak-anak. Pembelajaran bisa kita dapatkan dari mana saja, terutama dari sebuah permainan.

Menurut (Hasugian, 2020) mengatakan bahwa metode pembelajaran dapat diberikan secara berkala kepada anak dengan memberikan beberapa objek gambar dengan cara di acak dan setelah itu anak-anak harus menemukan objek gambar yang sama atau cocok. Menggunakan metode yang biasa dengan buku atau kertas bergambar anak-anak akan merasa cepat bosan karena metode itu sudah lama digunakan dan diulang beberapa kali. Hal ini dikarenakan *video game* edukatif dapat diterapkan sesuai dengan karakteristik anak usia dini terutama bagi siswa taman kanak-kanak yang suka bermain dan belajar, serta berimajinasi (Wijayanto & Siradj, 2017).

Algoritma *Fisher Yates Shuffle* adalah sebuah algoritma yang biasanya menggunakan masukan acak dengan harapan mencapai kinerja yang baik (Sharma & Garg, 2011). Algoritma tersebut berguna pada saat awal game dimulai, yaitu program akan mengacak *sprites/gambar* pada objek yang akan ditampilkan pada tampilan depan kartu dan memberikan tantangan kepada pemain harus mengingat letak kartu yang telah disediakan. Algoritma digunakan untuk mendapat hasil pengacakan yang cepat dan juga menghasilkan *output* yang bervariasi. Dengan penjelasan tersebut bisa menyimpulkan bahwa *Fisher Yates Shuffle algorithm* dapat mengacak data tanpa pengacakan berulang. Jika pertanyaannya sudah menghasilkan, maka tidak perlu memproduksinya lagi dengan metode menukar *indeks* yang dipilih (Revano et al., 2019).

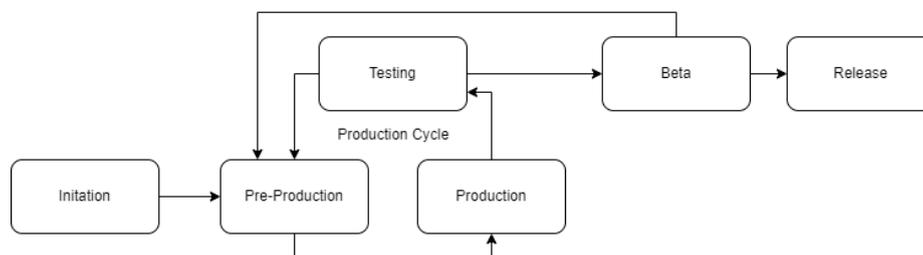
Game edukasi yang dibuat merupakan sarana media pendidikan menggunakan teknologi multimedia interaktif. Pada tema *game* bertujuan untuk mengenalkan kepada pemain tentang macam-macam huruf hijaiyah. Pemain dapat mengenal tentang huruf hijaiyah dengan mengingat huruf yang ditampilkan. Pemain dapat sekaligus menghafal huruf-huruf hijaiyah, karena pada dasarnya jika kita harus menebak kesamaan suatu gambar pada objek yang akan ditampilkan pada permainan maka pemain juga harus sekaligus mengingat dan melafalkan objek tersebut. Aplikasi dibuat menggunakan *game engine unity* dengan *platform android*. Pada aplikasi permainan *memory card game* ini tersedia banyak fitur yang membantu game agar berjalan lebih baik dan juga ada beberapa fitur tantangan yang akan diberikan. *Fitur* yang diberikan juga melatih pemain dalam cepat tanggap melakukan tugas yang diberikan. Seperti contohnya pada *game* tersedia *timer*, *turns*, dan juga *score*

pada setiap game yang berlangsung. Pada fitur tersebut telah disesuaikan pada level masing-masing pada saat kita memilih tingkat kesulitan dengan itu dapat mendukung pemain dalam melatih daya ingat dan konsentrasi pada saat bermain. Pada *BeginMenu* juga tersedia *panel* yang menjelaskan apa target yang harus diselesaikan berupa tata cara bermain dan juga penjelasan tentang huruf hijaiyah.

Pada pengujian aplikasi dilakukan pada proses pengacakan pada kartu yang ada pada permainan. Pengujian dilakukan untuk melihat pengacakan dari setiap level dapat menghasilkan kartu yang acak.

METODE PENELITIAN

Metode *Game Development Life Cycle* (GDLC) diusulkan untuk mengambil pendekatan berulang untuk memungkinkan tingkat yang lebih tinggi dari *fleksibilitas* terhadap perubahan selama proses pengembangan *video game*. (Mustofa et al., 2021). Ada 6 tahapan yang dimana masing-masing tahapan terkait satu sama lainnya dan terstruktur, tetapi penelitian ini hanya sampai tahap *beta*, tahapan GDLC terdapat pada Gambar 1 :



Gambar 1: Diagram *Game Development Life Cycle*

1. *Initiation*, merupakan tahapan ini merupakan pengumpulan ide berupa rancangan sebuah game edukasi untuk anak-anak dengan konsep *game jumble* bertemakan huruf hijaiyah. Pada tahapan ini mengumpulkan konsep kasar permainan yang nanti akan di implementasikan pada aplikasi. *Game jumble* edukasi mempunyai target pemain anak-anak untuk mempelajari huruf hijaiyah dan melatih daya ingat.
2. *Pre-production*, merupakan tahapan utama dari proses produksi melanjutkan pembuatan rancangan *game* yang berfokus pada pembuatan alur jalannya permainan menggunakan *flowchart* dan membuat *prototype* untuk menentukan *gameplay* yang akan diterapkan. Aplikasi game dibuat menggunakan *game engine Unity* dan menggunakan bahasa pemrograman C#.
3. *Production*, merupakan tahapan implementasi dari ide-ide yang dilakukan sebelumnya

dan disempurnakan seperti penambahan *asset* untuk menyempurnakan detail tampilan pada *game*. Pada tahap *production* berfokus pada penyempurnaan dari desain *prototype* pertama dari segi desain dan fungsi yang tersedia untuk mendapatkan hasil yang maksimal.

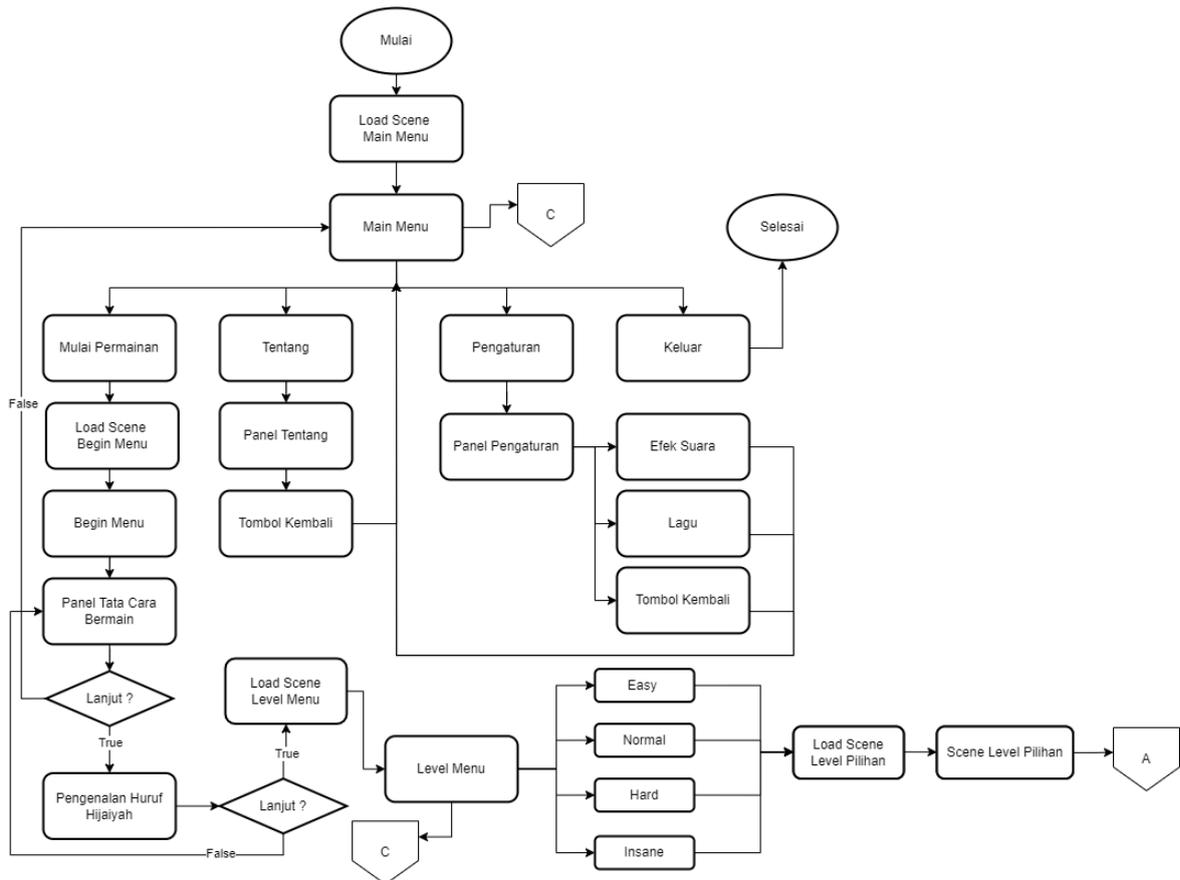
4. *Testing*, pada tahapan ini di uji proses pengacakan pada setiap level yang ada pada permainan. Pengujian bertujuan untuk melihat hasil yang didapatkan pada setiap pengujian mendapatkan hasil kartu yang berbeda pada setiap *game* dimulai.
5. *Beta*, merupakan tahapan dimana permainan akan diterima pihak ketiga untuk dilakukan *beta testing*. Pengujian bertujuan untuk menguji *game* apakah aplikasi berjalan lancar tanpa ada *error* atau *bug*, dan jika ditemukan *bug* maka akan mengulang tahapan *production cycle* untuk memperbaiki *bug* pada *game*. Pada pengujian mengumpulkan data hasil kuesioner dari para *reviewer* aplikasi yang telah dibuat pada *Google Form* sebanyak 10 pertanyaan pada *game* untuk melihat apakah fungsi pada aplikasi berjalan sesuai yang di harapkan selama pengujian.
6. *Release*, merupakan tahap terakhir pada *Game Development Life Cycle* (GDLC). Pada Tahap ini merupakan bentuk *final* dari aplikasi yang sudah dikembangkan yang berarti *game* sudah lulus pada proses *beta testing* dan sudah siap di terima publik.

HASIL DAN PEMBAHASAN

Perancangan Sistem

1. Flowchart Menu

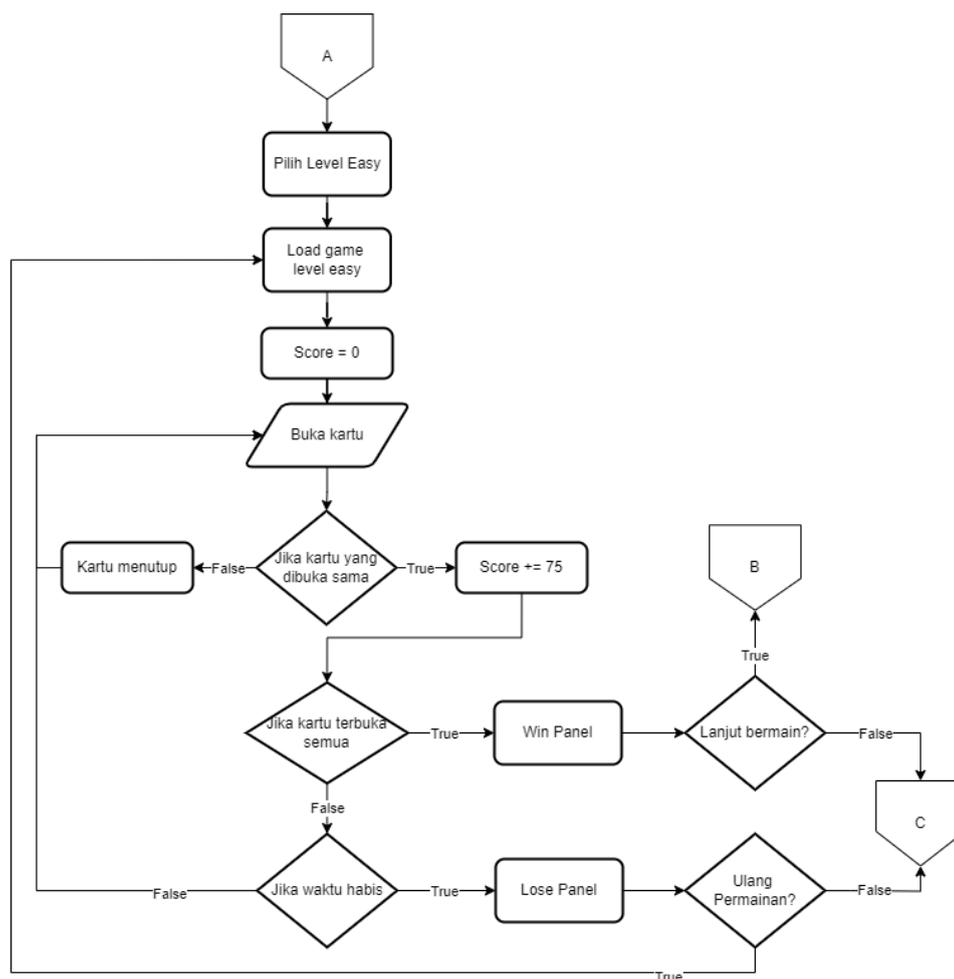
Pada Gambar 2, adalah tahapan alur pada menu, dimulai dari *start menu*, didalam tampilan utama terdapat pilihan mulai, pengaturan, tentang dan keluar. Pada saat memilih mulai akan berpindah ke *scene begin menu* yang berisi panel tata cara bermain dan pengenalan huruf hijaiyah. Pada menu *level* pemain memilih tingkat kesulitan dan berpindah ke *scene* bermain yang dipilih.



Gambar 2: Flowchart Menu

2. Flowchart Game

Pada Gambar 3, merupakan tahapan dari *game* yang di lanjutkan dari flowchart Gambar 2. Pada tahapan ini dimulai dari tahap pemilihan *level* dan pada awal mulai *game score* pemain dimulai dari 0. Pada *flowchart* menunjukkan bahwa jika kartu terbuka dan kartu *match* maka bertambah *score* dan jika salah maka kartu akan tertutup, jika kartu terbuka semua maka akan muncul *win panel*, jika tidak terbuka semua dan waktu habis maka akan muncul *lose panel*. Pada saat pemain menang maka akan ada pilihan lanjut bermain dan kembali ke *main menu*. Jika pemain kalah maka akan ada pilihan ulang permainan atau kembali ke *main menu*.



Gambar 3: Flowchart Game

Implementasi Algoritma Pengacakan Pada Aplikasi

Algoritma *Fisher-Yates Shuffle* digunakan karena algoritma ini lebih baik untuk melakukan metode pengacakan atau juga bisa bertujuan untuk metode pengacakan angka, dengan waktu eksekusi metode yang tidak lama serta tidak memakan waktu pada saat melakukan suatu pengacakan (Jefri nardi, 2019). Manfaat dari algoritma *shuffle Fisher Yates* adalah kecepatan dan akurasinya yang unggul dalam memperkirakan keacakan informasi (Gulappagol & Kumar, 2020). Metode pengacakan pada game terlihat seperti seperti *shuffle* dek pada kartu, dimana pada proses pengacakan semua kartu sehingga susunannya kartu menjadi teracak. Pada penelitian kali ini menggunakan algoritma *fisher yates modern method*. Metode *modern* dipilih karena metode ini memang khusus digunakan untuk pengacakan dengan sistem komputerisasi dan hasil pengacakan bisa lebih variatif (Santoso & Gunawan, 2021). Berikut merupakan alur pada pengacakan kartu yaitu (Ayu Irawati et al., 2017) :

1. Input indeks kartu dari 1 hingga N yang diinginkan.

2. Pilih indeks secara acak.
3. Tukar indeks (x) dengan indeks terakhir pada (n) yang belum terpilih sebelumnya.
4. Ulangi langkah 2 sampai 3 kali sampai semua angka sudah tercoret.
5. Hasil perhitungan merupakan pemutasian acak dari angka awal.

Berikut merupakan tabel perhitungan algoritma *Fisher-Yates Shuffle Modern Method* pada *game jumble* hijaiyah :

Tabel 1. Tabel Perhitungan *Algoritma Fisher-Yates Shuffle Modern Method*

No	Range	Roll	Scratch	Result
1.	1 - 30	26	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28,29,30	26
2.	1 - 29	24	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24
3.	1 - 28	10	1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24,10
4.	1 - 27	11	1,2,3,4,5,6,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24,10,12
5.	1 - 26	10	1,2,3,4,5,6,7,8,9,13,14,15,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24,10,12,11
6.	1 - 25	5	1,2,3,4,6,7,8,9,13,14,15,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5
7.	1 - 24	8	1,2,3,4,6,7,8,13,14,15,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9
8.	1 - 23	5	1,2,3,4,7,8,13,14,15,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6
9.	1 - 22	9	1,2,3,4,7,8,13,14,16,17,18,19,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15
10.	1 - 21	12	1,2,3,4,7,8,13,14,16,17,18,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19
11.	1 - 20	6	1,2,3,4,7,13,14,16,17,18,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19,8
12.	1 - 19	7	1,2,3,4,7,13,16,17,18,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19,8,14
13.	1 - 18	5	1,2,3,4,13,16,17,18,20,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19,8,14,7
14.	1 - 17	9	1,2,3,4,13,16,17,18,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19,8,14,7,9
15.	1 - 16	8	1,2,3,4,13,16,17,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19,8,14,7,9,18
16.	1 - 15	3	1,2,4,13,16,17,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3
17.	1 - 14	5	1,2,4,13,17,21,22,23,25,27,28,29,30	26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16
18.	1 - 13	13	1,2,4,13,17,21,22,23,25,27,28,29	26,24, 10,12,11,5,9,6,15,19,8,14,7,9,18,3,16,30
19.	1 - 12	10	1,2,4,13,17,21,22,23,25,28,29	26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16, 30,27
20.	1 - 11	7	1,2,4,13,17,21,23,25,28,29	26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16, 30,27,22
21.	1 - 10	4	1,2,4,17,21,23,25,28,29	26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16, 30,27,22,13
22.	1 - 9	6	1,2,4,17,21,25,28,29	26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16

					,30,27,22,13,23
23.	1 - 8	8	1,2,4,17,21,25,28		26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16 ,30,27,22,13,23,29
24.	1 - 7	5	1,2,4,17,25,28		26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16 ,30,27,22,13,23,29,21
25.	1 - 6	2	1,4,17,25,28		26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16 ,30,27,22,13,23,29,21,2
26.	1 - 5	3	1,4,25,28		26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16 ,30,27,22,13,23,29,21,2,17
27.	1 - 4	4	1,4,25		26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16 ,30,27,22,13,23,29,21,2,17,28
28.	1 - 3	3	1,4		26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16 ,30,27,22,13,23,29,21,2,17,28,25
29.	1 - 2	1	4		26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3,16 ,30,27,22,13,23,29,21,2,17,28,25,1
Hasil Perhitungan :					4,26,24,10,12,11,5,9,6,15,19,8,14,7,9,18,3, 16,30,27, 22,13,23,29,21,2,17,28,25,1

Pada Tabel 1 merupakan contoh pengujian perhitungan pada pengacakan kartu yang berjumlah 30 *sprites* sesuai dengan jumlah huruf pada hijaiyah. *Range* merupakan angka jumlah yang belum terseleksi. *Roll* merupakan urutan angka acak yang terpilih. *Scratch* merupakan kumpulan angka yang belum dipilih. *Result* merupakan hasil yang didapat dari permutasi acak (Arviansyah et al., 2020). Pada setiap level yang ada memiliki *rules* tersendiri, seperti pada waktu permainan dan juga jumlah kartu yang ada pada saat game dimulai. Hasil dari pengacakan di sesuaikan dengan jumlah kartu yang ada pada setiap level. Berikut *rules* bermain pada setiap level permainan :

Tabel 2. Tabel Level Pada Permainan

Level	Rules	Output
Easy	4x3 (6 Match)	12 Kartu
Normal	5x4 (10Match)	20 Kartu
Hard	6x4 (12 Match)	24 Kartu
Very Hard	8x4 (16 Match)	32 Kartu

Pada Tabel 2 merupakan rancangan level yang ada pada permainan. *Level* merupakan tingkat kesulitan pada permainan. *Rules* merupakan jumlah baris dan kolom yang ada pada board dan jumlah kartu yang match. *Output* merupakan jumlah kartu yang akan *spawn* pada *board*. Terdapat pengujian algoritma pada setiap level , Berikut merupakan hasil dari pengujiannya :

Tabel 3. Tabel Hasil Pengujian

No	Level																			
	Easy					Normal					Hard					Very Hard				
	Pengujian																			
	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
1	3	25	19	3	2	2	23	1	2	3	3	23	29	23	9	12	14	30	10	11
2	12	13	21	4	14	24	15	23	28	28	25	12	30	1	30	13	21	1	3	2
3	1	26	15	21	23	14	14	21	15	2	12	6	14	16	29	2	25	23	4	3
4	26	23	22	24	18	15	19	12	14	16	14	7	22	15	15	1	26	17	12	4
5	25	14	3	14	19	25	12	8	22	14	21	3	24	5	3	4	27	18	30	15
6	14	17	1	2	1	26	13	15	27	9	28	2	8	8	5	18	20	2	16	25
7						17	1	10	9	5	18	4	6	4	17	15	24	5	11	27
8						18	3	28	5	22	19	24	1	18	16	19	12	21	17	16
9						12	4	7	23	19	26	27	9	28	19	28	4	3	26	28
10						19	5	26	7	13	4	13	13	17	26	29	16	4	7	8
11											13	1	27	14	1	30	8	9	2	9
12											1	5	12	19	18	3	3	14	28	10
13																9	13	12	21	12
14																25	11	15	20	18
15																26	10	19	19	29
16																24	9	10	22	1

Pada Tabel 3 merupakan pengujian pada setiap level dengan menguji jumlah kartu yang *match* pada masing-masing level dan dilihat apakah jumlah data yang dihasilkan berbeda pada setiap pengujian. Pada setiap level dilakukan pengujian sejumlah 5 kali untuk mendapatkan hasil yang akurat. Dapat dilihat hasil dari pengujian bahwa algoritma *fisher-yates shuffle* memiliki kelebihan yaitu memiliki kesederhanaan dalam metode pengacakannya serta kompleksitas algoritma yang optimal (Zuliani et al., 2018).

Implementasi Sistem

Pada aplikasi permainan jumble hijaiyah memiliki beberapa scene yang dimana memiliki desain UI masing-masing. Berikut merupakan hasil berupa pada memory card game :

1. Tampilan Desain Main Menu

Halaman ini merupakan tampilan utama pada saat membuka *game*. Gambar 4 berisi *button* yaitu *Play*, *About*, *Setting* dan *Quit*.



Gambar 4: Tampilan Desain Start Menu

2. Tampilan Desain Panel Settings

Halaman ini berfungsi untuk mengatur *sound effect* dan *music* pada saat permainan berlangsung



Gambar 5: Tampilan Desain Panel Setting

3. Tampilan Menu Tata Cara Bermain dan Pengenalan Huruf Hijaiyah

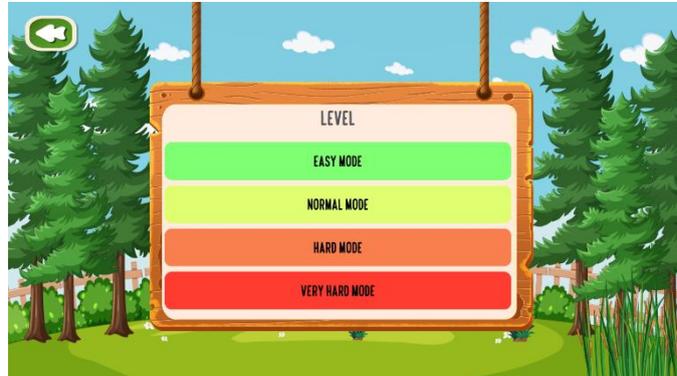
Pada *scene begin menu* ini berisi tentang tata cara bermain dan pengenalan huruf hijaiyah agar pemain dapat memahami alur dalam permainan beserta pengenalan huruf hijaiyah. Pada panel memiliki penjelasan berupa gambar dan terdapat penjelasan berupa suara.



Gambar 6: Tampilan Desain Menu Tata Cara Bermain dan Pengenalan Huruf Hijaiyah

4. Tampilan Menu Level

Halaman ini pemain bisa memilih tingkat kesulitan pada saat bermain. Setiap tingkatan memiliki *rules* masing-masing sesuai level yang dipilih.



Gambar 7: Tampilan Desain Menu Level

5. Tampilan Desain Bermain Scene Easy Mode

Setelah pemain memilih kategori easy maka akan keluar jumlah kartu 4x3 dengan waktu 60 detik.



Gambar 8: Tampilan Desain Bermain Scene Easy Mode

6. Tampilan Desain Menu *Panel Win* dan *Lose*

Tampilan *win panel* dan *lose panel* akan keluar pada saat pemain menyelesaikan permainan. Tampilan pada saat kalah mempunyai desain yang sama, yang membedakan hanya tulisan diubah menjadi *game over*.



Gambar 9: Tampilan Desain Menu Panel Win

SIMPULAN

Setelah dilakukan implementasi dan pengujian pada pembuatan *video game* edukasi *jumble* dapat ditarik kesimpulan bahwa:

1. Implementasi algoritma *Fisher Yates Shuffle* dapat diterapkan pada proses pengacakan di game edukasi *jumble* bertema hijaiyah
2. Pada proses pengujian algoritma pengacakan mendapat hasil yang baik yang menghasilkan urutan yang bervariasi dan tidak berulang.
3. Implementasi dari metode *Game Development Life Cycle* berjalan dengan baik untuk mendeskripsikan tahapan-tahapan pengembangan pada aplikasi.

REFERENSI

- Arviansyah, Y., Nurfaizah, N., & Waluyo, R. (2020). Penerapan Algoritma Fisher Yates Shuffle Pada Aplikasi TOEFL Preparation Berbasis Web. *Jurnal Buana Informatika*, 11(2), 111. <https://doi.org/10.24002/jbi.v11i2.3622>
- Ayu Irawati, D., Lugman Affandi, dan, Nur Rahmanto, A., Teknologi Inforasi, J., & Negeri Malang, P. (2017). *Prosiding SENTIA 2017-Politeknik Negeri Malang PENERAPAN ALGORITMA FISHER-YATES PADA PENGACAKAN SOAL GAME ARITMATIKA*. 9, 101–106.
- Gulappagol, L., & Kumar, K. B. S. (2020). *Encryption in Transform Domain Video Steganography*. 13(13), 52–57.
- Hasugian, A. H. (2020). *Perancangan Memory Game dengan Menggunakan Unreal Engine*. 1(1), 1–6.
- Jefri nardi, A. (2019). Berbasis Unity 3D. *Jurnal Algor*, x, 1–11.
- Mustofa, Lasmana Putra, J., & Kesuma, C. (2021). Penerapan Game Development Life Cycle

- Untuk Video Game Dengan Model Role Playing Game. *Computer Science (CO-SCIENCE)*, 1(1), 27–34. <http://jurnal.bsi.ac.id/index.php/co-science>
- Revano, T. F., Garcia, M. B., Habal, B. G. M., Contreras, J. O., & Enriquez, J. B. R. (2019). Logical guessing riddle mobile gaming application utilizing fisher yates algorithm. *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018*, 1–4. <https://doi.org/10.1109/HNICEM.2018.8666302>
- Santoso, A., & Gunawan, W. (2021). Implementasi Algoritma Fisher-Yates Shuffle Dan Fuzzy Tsukamoto Pada Aplikasi Pembelajaran Pemrograman Dasar Berbasis *Hexagon Jurnal Teknik dan Sains*, 2, 63–72. <http://www.jurnal.uts.ac.id/index.php/hexagon/article/view/879>
- Sharma, K., & Garg, D. (2011). Randomized Algorithms: Methods and Techniques. *International Journal of Computer Applications*, 28(11), 29–32. <https://doi.org/10.5120/3436-4510>
- Wijaya, A., & Apridiansyah, Y. (2020). Penerapan Algoritma Fisher Yates Shuffle Pada Media Pembelajaran Mapel Agama Islam Berbasis Android. *Jurnal Informatika Upgris*, 6(1). <https://doi.org/10.26877/jiu.v6i1.5747>
- Wijayanto, P. W., & Siradj, Y. (2017). The Educational Game “Indonesian Tribes” for the Kindergarten Students. *IJPTE : International Journal of Pedagogy and Teacher Education*, 1(1), 27–36. <https://doi.org/10.20961/ijpte.v1i1.8456>
- Zuliani, S. A., Winarno, E., & Studi. (2018). Game Pembelajaran Membaca Iqra ' Menggunakan Metode Fisher Yates Shuffle. *Prosiding SINTAK*, 2, 333–337. <https://www.unisbank.ac.id/ojs/index.php/sintak/article/view/6639>