

# Perancangan *Game* Edukasi *Python* Berbasis Web Menggunakan Algoritma *Knuth Shuffle*

<sup>1\*</sup>Mukhlifatus Shodikin, <sup>2</sup>Danang Wahyu Widodo, <sup>3</sup>Risky Aswi Ramadhani

<sup>1,2,3</sup> Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: \*<sup>1</sup>[muklissodik86@gmail.com](mailto:muklissodik86@gmail.com), <sup>2</sup>[danangwahyuwidodo@unpkediri.ac.id](mailto:danangwahyuwidodo@unpkediri.ac.id),  
<sup>3</sup>[riskyaswiramadhani@gmail.com](mailto:riskyaswiramadhani@gmail.com)

*Penulis Korespondens : Mukhlifatus Shodikin*

**Abstrak**—Media pembelajaran interaktif sangat dibutuhkan dalam pembelajaran pemrograman di SMK. Meskipun *Python* populer dan mudah dipahami, banyak siswa masih kesulitan dalam memahami konsep dasarnya. Penelitian ini merancang *EduPython Adventure*, *game* edukasi berbasis web yang menggunakan algoritma *Knuth Shuffle* untuk mengacak soal sehingga tidak repetitif. Metode yang digunakan adalah *Game Development Life Cycle (GDLC)* yang meliputi analisis kebutuhan, perancangan, pengembangan *gameplay*, dan implementasi algoritma. Hasil implementasi menunjukkan bahwa *game* dapat menyajikan soal secara acak dan adaptif. Pengujian algoritma menunjukkan distribusi acak yang merata. *Game* ini dapat diakses tanpa instalasi, mendukung berbagai perangkat, dan menyediakan fitur pembelajaran bertahap. Kesimpulannya, *game* ini berpotensi menjadi media pembelajaran yang menarik dan inovatif untuk mendukung pemahaman dasar *Python* di SMK.

**Kata Kunci**— *EduPython*, *GDLC*, *game* edukasi, *Knuth Shuffle*, *Python*

**Abstract**— *Interactive learning media are essential in programming education at vocational schools. Although Python is popular and easy to understand, many students still struggle with its basic concepts. This study designed EduPython Adventure, a web-based educational game using the Knuth Shuffle algorithm to randomize questions and prevent repetition. The development method applies the Game Development Life Cycle (GDLC), covering needs analysis, design, gameplay, and algorithm implementation. The implementation results show that the game presents randomized and adaptive questions. Shuffle testing indicates a fair distribution. The game is accessible without installation, supports various devices, and provides a progressive learning system. In conclusion, this game has the potential to be an innovative learning tool to support Python comprehension at vocational schools.*

**Keywords**— *EduPython*, *educational game*, *GDLC*, *Knuth Shuffle*, *Python*

This is an open access article under the CC BY-SA License.



## I. PENDAHULUAN

Pemrograman telah menjadi salah satu kompetensi penting di era digital, terutama dalam pendidikan kejuruan seperti Sekolah Menengah Kejuruan (SMK). Penguasaan pemrograman tidak hanya meningkatkan kemampuan teknis, tetapi juga mendorong pengembangan keterampilan berpikir logis, analitis, dan pemecahan masalah yang sangat dibutuhkan di dunia kerja modern [2], [3], [10]. Salah satu bahasa pemrograman yang banyak digunakan untuk pemula adalah *Python* karena memiliki sintaks yang sederhana, mudah dipahami, dan

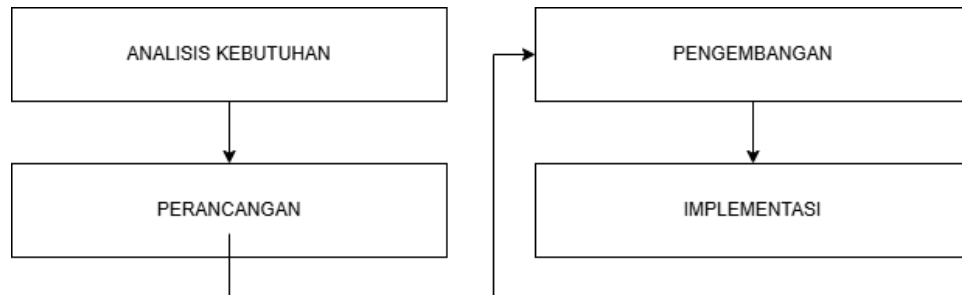
mendukung berbagai paradigma pemrograman [3]. Namun, dalam praktiknya, banyak siswa SMK masih mengalami kesulitan dalam memahami dasar-dasar *Python*, terutama dalam menyusun logika algoritmik dan urutan instruksi yang tepat [10]. Kondisi ini diperparah dengan minimnya media pembelajaran yang interaktif dan menyenangkan, sehingga proses belajar sering kali monoton dan kurang memotivasi [1], [9].

Berbagai studi menunjukkan bahwa pendekatan pembelajaran berbasis *game* (*game-based learning*) memiliki potensi besar dalam meningkatkan motivasi, keterlibatan, dan efektivitas belajar siswa, khususnya dalam bidang ilmu komputer [1], [3], [9]. Dalam konteks ini, integrasi *Python* dalam pembelajaran berbasis proyek dan kurikulum Merdeka Belajar juga telah terbukti mampu meningkatkan hasil akademik siswa secara signifikan [2]. Salah satu elemen penting dalam *game* edukasi adalah adanya tantangan yang bervariasi dan tidak repetitif, yang dapat dicapai melalui penggunaan algoritma pengacakan soal. Algoritma *Knuth Shuffle*, atau yang dikenal juga sebagai *Fisher-Yates Shuffle*, merupakan metode populer dalam menghasilkan urutan acak secara efisien dan adil [4], [6]. Beberapa penelitian telah menerapkan algoritma ini dalam konteks pembelajaran seperti kuis berbasis lingkungan [8], aplikasi *m-learning* [6], hingga materi keagamaan [5], dan menunjukkan hasil yang baik dalam hal efektivitas dan efisiensi. Saokani et al. [4] bahkan membandingkan performa *Knuth Shuffle* dengan algoritma *Linear Congruent* dan menemukan bahwa *Knuth Shuffle* lebih unggul dalam menghasilkan variasi soal yang tidak mudah ditebak. Selain itu, riset oleh Hooshyar et al. [7] menunjukkan bahwa variasi tantangan dalam media belajar dapat mendorong keterlibatan dan mengurangi perilaku menunda-nunda belajar (*procrastination*) pada siswa.

Kendati demikian, penerapan algoritma *Knuth Shuffle* secara khusus dalam *game* edukasi *Python* untuk siswa SMK masih jarang ditemukan. Penelitian ini hadir untuk mengisi kesenjangan tersebut, dengan tujuan utama (1) mengimplementasikan algoritma *Knuth Shuffle* dalam sistem pengacakan soal pada *game* edukasi, serta (2) merancang *game* edukasi berbasis web yang interaktif untuk membantu siswa SMK memahami konsep dasar *Python* dengan cara yang lebih menarik dan adaptif. Dengan pendekatan sistematis berbasis *Game Development Life Cycle* (GDLC), diharapkan hasil dari penelitian ini tidak hanya memberi kontribusi praktis dalam pengembangan media pembelajaran inovatif, tetapi juga kontribusi teoretis dalam integrasi algoritma dalam konteks pendidikan digital yang responsif terhadap kebutuhan siswa SMK masa kini.

## II. METODE

Penelitian ini menggunakan pendekatan *Game Development Life Cycle* (GDLC) yang bertujuan mendeskripsikan semua elemen desain dan implementasi *game* edukasi berbasis web berjudul *EduPython Adventure*. Pendekatan ini terbagi dalam enam tahap, dengan penjelasan dan visual sebagai berikut:



Gambar 1. Diagram GDLC

### A. Analisis Kebutuhan

Analisis kebutuhan dilakukan sebagai tahap awal dalam proses perancangan *game* edukasi berbasis web yang bertujuan untuk meningkatkan pemahaman siswa SMK terhadap dasar-dasar pemrograman *Python*. Aktivitas analisis ini mencakup identifikasi kebutuhan fungsional seperti mekanisme pengacakan soal menggunakan algoritma *Knuth Shuffle* dan desain *gameplay* interaktif yang mendukung pembelajaran aktif. Selain itu, juga diidentifikasi kebutuhan non-fungsional meliputi kompatibilitas lintas perangkat, performa sistem yang optimal, serta kemudahan akses melalui platform web. Pengumpulan data dilakukan melalui studi literatur dan analisis kurikulum untuk memahami karakteristik target pengguna, yaitu siswa SMK tingkat pemula yang menjadi sasaran utama dalam pengembangan media pembelajaran ini.

### B. Perancangan Game

*Game* edukasi *EduPython Adventure* mengusung konsep petualangan interaktif dalam dunia digital yang kacau akibat serangan karakter antagonis bernama *Bug*, simbol dari kesalahan dalam pemrograman. Pemain berperan sebagai *Python*, seorang karakter *programmer* muda yang ditugaskan oleh AI Mentor untuk memulihkan kestabilan sistem. Cerita ini mengalir melalui dialog naratif yang muncul di setiap fase permainan, memberikan arahan, tantangan, serta motivasi kepada pemain. Dunia permainan dibagi menjadi beberapa wilayah dengan tema spesifik yang merepresentasikan materi *Python*, seperti *Data Valley*, *Logic Mountain*, dan *Algorithm Forest*. Masing-masing area memiliki tantangan yang berkaitan langsung dengan kompetensi dasar pemrograman *Python*. Dalam aspek *gameplay*, pemain diminta memilih tingkat kesulitan terlebih dahulu, yaitu Pemula, Menengah, atau Lanjutan, yang menentukan kompleksitas soal dan tantangan di setiap level. Level Pemula membahas variabel, tipe data, dan input/output sederhana, sedangkan level Menengah mencakup logika percabangan dan perulangan. Level Lanjutan memfokuskan pada fungsi, *array*, dan algoritma sederhana. Setiap level terdiri dari 10 *stage*, dan setiap *stage* memuat dua soal berbeda, sehingga satu level memiliki 20 soal yang disajikan secara acak agar tidak repetitif.

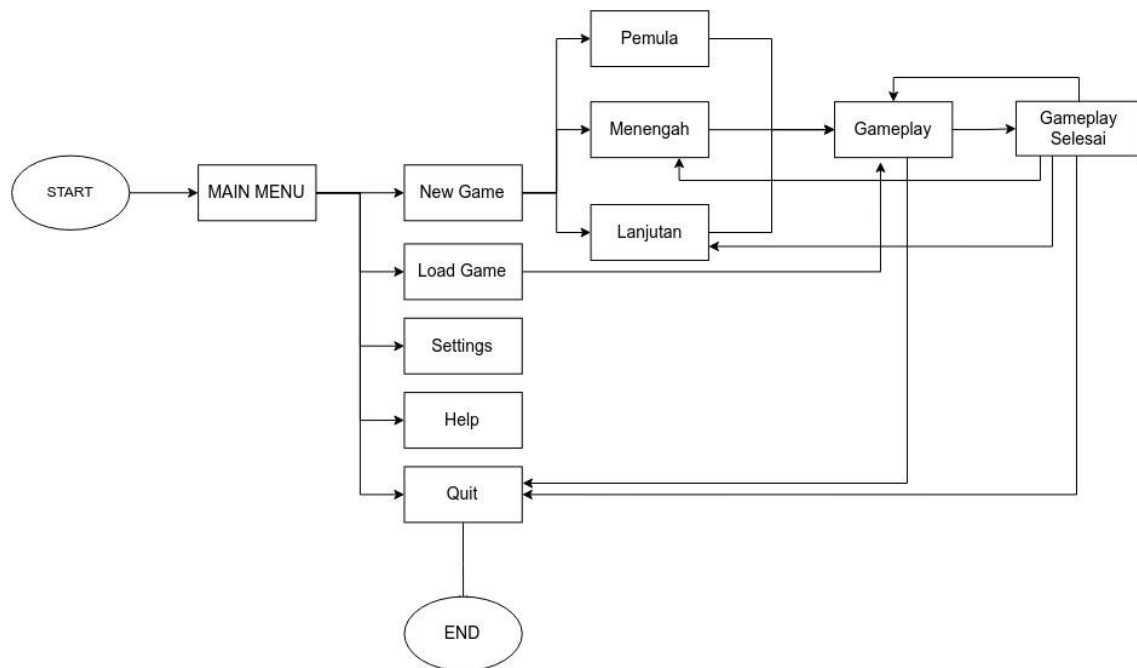
Sistem soal dalam *game* ini didukung oleh algoritma *Knuth Shuffle*, yang diterapkan untuk mengacak urutan soal dan opsi jawaban setiap kali permainan dimulai. Hal ini bertujuan agar pemain tidak menghafal jawaban dan lebih fokus pada pemahaman konsep. Tantangan yang diberikan mencakup pengisian kode, penyusunan ulang baris program, dan perbaikan *bug* logika. Pemain diberi skor berdasarkan akurasi, efisiensi kode, dan kecepatan menjawab. Sistem juga menyediakan *hint* terbatas sebagai bantuan, namun penggunaannya akan mengurangi poin.

Secara teknis, *game* ini dirancang untuk dijalankan di browser pada perangkat desktop. Aplikasi dibangun menggunakan kombinasi teknologi HTML, CSS, *JavaScript* untuk *frontend*, serta *Python Flask* sebagai *backend*. Data pemain dan progres permainan dikelola menggunakan SQLite. Desain grafis dikembangkan dengan perangkat lunak seperti Figma, Photoshop, dan Illustrator. Untuk memastikan pengalaman bermain yang optimal, dibutuhkan perangkat dengan spesifikasi minimal prosesor Intel Core i3, RAM 4GB, dan koneksi internet stabil minimal 1 Mbps.

Dengan pendekatan berbasis cerita dan struktur level bertahap yang dipadukan dengan mekanik soal acak dan visual menarik, *EduPython Adventure* tidak hanya menghadirkan pengalaman belajar yang menyenangkan, tetapi juga efektif dalam membangun pemahaman siswa terhadap logika pemrograman *Python* secara kontekstual dan berulang.

### C. *Screenflow* dan Diagram Navigasi

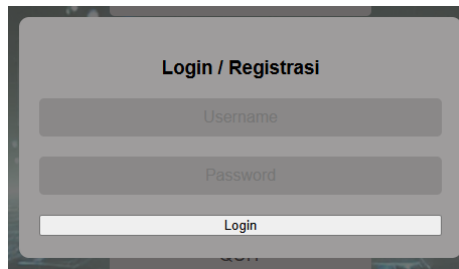
Diagram *screenflow* pada Gambar 1 menunjukkan alur logika navigasi antar layar.



Gambar 2. Diagram *Screenflow* Game *EduPython Adventure*

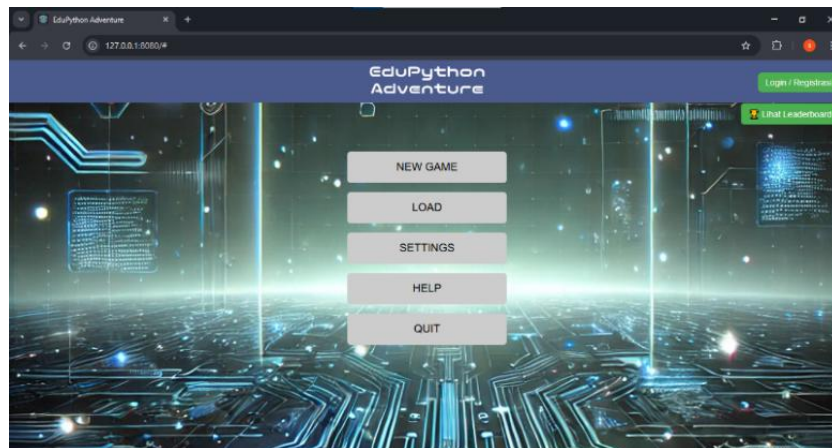
Gambar ini menggambarkan alur logika navigasi antar layar pada *game EduPython Adventure* secara keseluruhan. Mulai dari tampilan awal (*start screen*), pemain diarahkan ke menu utama, kemudian memilih menu *gameplay* untuk memulai permainan. Selama permainan berlangsung, pemain dapat sewaktu-waktu melakukan pause, menyimpan progres, atau keluar dari permainan. Alur ini dibuat fleksibel agar dapat menyesuaikan kebutuhan pengguna pada setiap level dan mendorong pengalaman bermain yang intuitif dan terstruktur.

#### D. Desain Antarmuka Pengguna (UI)



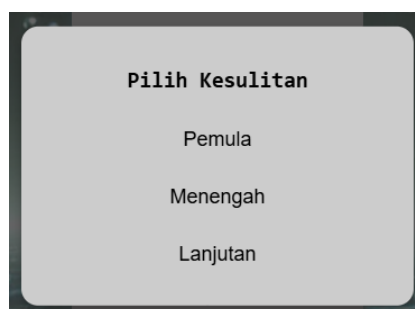
Gambar 3. Antarmuka *Login*

Antarmuka *login* menampilkan form input untuk *Username* dan *Password*, yang menjadi akses utama bagi pemain ke dalam sistem *game*. *Login* diperlukan untuk mengidentifikasi pemain secara unik dan memungkinkan sistem menyimpan serta memuat progres permainan sebelumnya. Fitur ini juga mendukung manajemen pengguna, seperti *leaderboard* dan personalisasi pengalaman bermain berdasarkan progres.



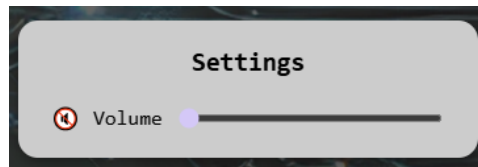
Gambar 4. Menu Utama

Tampilan ini merupakan pusat navigasi dalam *game* yang menyediakan akses ke berbagai fungsi utama, antara lain *New Game* untuk memulai permainan baru, *Load* untuk memuat permainan sebelumnya, *Settings* untuk mengatur preferensi permainan, *Help* untuk mendapatkan petunjuk penggunaan, dan *Quit* untuk keluar dari aplikasi. Desain menu utama dibuat sederhana namun informatif agar mudah dipahami oleh pengguna dari berbagai tingkat usia dan pengalaman.



Gambar 5. Pemilihan Level

Gambar ini memperlihatkan tampilan antarmuka saat pemain memilih tingkat kesulitan permainan: Pemula, Menengah, atau Lanjutan. Masing-masing tingkat memiliki kompleksitas soal dan tantangan yang berbeda. Level Pemula mencakup topik dasar seperti variabel dan *input/output*, Menengah fokus pada percabangan dan perulangan, sedangkan Lanjutan membahas fungsi, *array*, dan algoritma. Pemilihan level memungkinkan pembelajaran bertahap dan sesuai dengan kemampuan pemain.



Gambar 6. Menu Pengaturan (*Settings*)

Menu ini memberikan kontrol kepada pemain untuk menyesuaikan pengaturan audio *game*, seperti mengatur volume suara latar (*background music*). Fitur ini penting untuk meningkatkan kenyamanan bermain dan mengakomodasi preferensi individu, terutama dalam lingkungan pembelajaran yang mungkin memerlukan mode senyap.

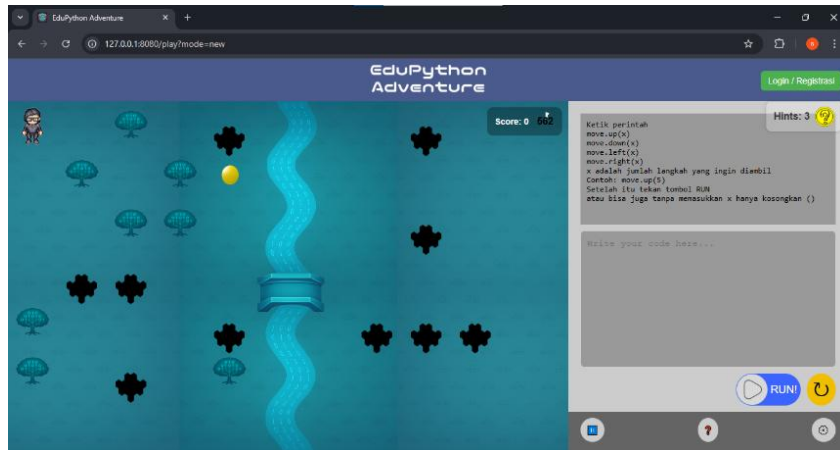
#### E. Dialog Narasi dan *Storytelling*



Gambar 7. Dialog Cerita Awal

Menampilkan bagian awal dari narasi *game* yang menjelaskan latar belakang cerita, tokoh utama, dan tujuan dari permainan. Dalam skenario ini, pemain berperan sebagai karakter *Python* yang ditugaskan oleh AI Mentor untuk memulihkan dunia digital yang terganggu oleh karakter jahat bernama *Bug*. Dialog ini tidak hanya menjadi pembuka, tetapi juga berfungsi sebagai pengantar pembelajaran agar pemain lebih terlibat secara emosional dan memahami konteks edukatif dalam *game*.

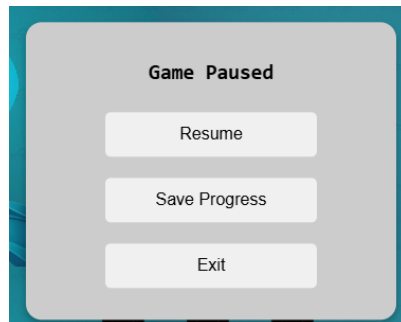
#### F. Antarmuka *Gameplay*



Gambar 8. *Gameplay EduPython*

Gambar ini menunjukkan antarmuka utama saat permainan berlangsung. Pemain mengontrol karakter menggunakan perintah berbasis kode *Python* seperti *move.up(x)* dan menyelesaikan tantangan yang muncul. Dalam layar ini terdapat editor kode, skor permainan, *timer*, sistem *hint*, dan tampilan level yang sedang dimainkan. *Gameplay* dirancang untuk mendorong interaksi langsung dengan logika pemrograman dalam situasi kontekstual yang menyenangkan.

#### G. Fitur Penyimpanan Progress



Gambar 9. Menu *Pause* dan *Save Progress*

Tampilan ini muncul ketika pemain menjeda permainan. Dalam menu ini terdapat opsi untuk melanjutkan permainan, menyimpan progres ke dalam basis data, atau keluar dari permainan. Fitur penyimpanan progres memungkinkan pemain untuk melanjutkan permainan di lain waktu tanpa kehilangan pencapaian sebelumnya. Ini penting untuk mendukung fleksibilitas waktu belajar dan menjaga motivasi pemain dalam menyelesaikan seluruh level.

#### H. Implementasi Algoritma *Knuth Shuffle*

Implementasi algoritma *Knuth Shuffle* (atau dikenal sebagai *Fisher-Yates Shuffle*) digunakan untuk melakukan pengacakan elemen dalam *array* secara efisien dan tidak berulang, guna mencegah prediktabilitas soal dalam permainan. Algoritma ini diimplementasikan dalam bahasa *JavaScript* dan digunakan untuk mengacak urutan soal serta opsi jawaban pada setiap sesi permainan.

Adapun implementasi algoritma ditulis sebagai berikut:

```
function shuffle(array) {
  for (let i = array.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [array[i], array[j]] = [array[j], array[i]];
  }
  return array;
}
```

Proses pengacakan ini dijalankan sebelum *game* dimulai agar setiap pemain mendapatkan susunan soal dan pilihan yang berbeda, sehingga mendorong pemahaman konsep dan bukan sekadar menghafal jawaban. Tabel berikut menggambarkan simulasi pengacakan berdasarkan *array* awal [A, B, C, D, E]:

Table 1. simulasi pengacakan

Iterasi	<i>i</i>	Nilai <i>j</i> = <i>random</i> (0, <i>i</i> )	Elemen yang Ditukar ( <i>array</i> [ <i>i</i> ] $\rightleftharpoons$ <i>array</i> [ <i>j</i> ])	Array Setelah Penukaran
1	4	2	E $\rightleftharpoons$ C	[A, B, E, D, C]
2	3	1	D $\rightleftharpoons$ B	[A, D, E, B, C]
3	2	0	E $\rightleftharpoons$ A	[E, D, A, B, C]
4	1	1	D $\rightleftharpoons$ D ( <i>no change</i> )	[E, D, A, B, C]

Dengan penerapan algoritma ini, sistem soal menjadi lebih adaptif dan dinamis, sejalan dengan prinsip variabilitas dalam desain *game* edukatif. Pengacakan ini terbukti mampu mengurangi kecenderungan pemain untuk mengingat urutan soal serta meningkatkan fokus pada pemahaman konsep pemrograman *Python*.

### III. HASIL DAN PEMBAHASAN

#### A. Implementasi Algoritma *Knuth Shuffle*

Penggunaan algoritma *Knuth Shuffle* dalam *game EduPython Adventure* bertujuan untuk mengacak urutan soal sehingga setiap sesi permainan menghasilkan pengalaman belajar yang berbeda. Implementasi algoritma dilakukan menggunakan bahasa *JavaScript* dan diterapkan secara langsung pada kumpulan soal yang diambil dari berkas *questions.json*.

Fungsi ini melakukan pertukaran elemen secara acak dari indeks terakhir hingga indeks pertama dengan elemen indeks acak yang berada di bawahnya. Dengan demikian, setiap soal memiliki peluang yang sama untuk muncul di posisi mana pun, yang mendukung prinsip keacakan merata.



## B. Pengujian Keacakan (*Functional Shuffle Testing*)

Pengujian dilakukan untuk memastikan bahwa algoritma *Knuth Shuffle* menghasilkan distribusi soal yang acak dan adil. Pengujian dilakukan terhadap 10 soal dari level *beginner* dengan cara menjalankan proses pengacakan sebanyak 1000 iterasi. Parameter yang diamati adalah frekuensi kemunculan tiap soal di posisi pertama dalam urutan hasil pengacakan.

Hasil distribusi dari 1000 kali pengacakan ditunjukkan pada Tabel 1 berikut:

Tabel 2. Distribusi Soal di Posisi Pertama

Soal ke-	Frekuensi Muncul	Persentase
1	108	10.8%
2	95	9.5%
3	102	10.2%
4	97	9.7%
5	101	10.1%
6	99	9.9%
7	96	9.6%
8	102	10.2%
9	97	9.7%
10	103	10.3%

## C. Analisis Hasil

Distribusi frekuensi pada Tabel 1 menunjukkan bahwa tidak terdapat dominasi soal tertentu dalam posisi pertama. Setiap soal memiliki kemungkinan kemunculan yang mendekati ideal, yaitu sekitar 10% dari 1000 kali pengacakan. Rentang distribusi berkisar antara 9.5% hingga 10.8%, yang menunjukkan bahwa algoritma telah bekerja secara adil dan acak sesuai ekspektasi. Tidak ditemukan pola yang dapat diprediksi dari hasil pengacakan.

## D. Evaluasi Sistem

Dari sisi efisiensi waktu, algoritma *Knuth Shuffle* menunjukkan kompleksitas waktu  $O(n)$  yang efisien. Dalam proses pengacakan 1000 kali terhadap 10 soal, tidak ditemukan keterlambatan atau hambatan kinerja pada perangkat dengan spesifikasi standar. Validasi keacakan menunjukkan hasil yang merata dan tidak bias, sementara dari sisi stabilitas implementasi, tidak ditemukan *error* baik pada perangkat desktop. Hal ini mengindikasikan bahwa algoritma *Knuth Shuffle* tidak hanya bekerja secara teoritis tetapi juga berhasil diterapkan secara praktis dalam sistem permainan berbasis web.

## IV. KESIMPULAN

Penelitian ini telah berhasil merancang *EduPython Adventure*, sebuah *game* edukasi berbasis web sebagai media pembelajaran interaktif untuk mendukung pemahaman dasar *Python* bagi siswa SMK. Pengembangan dilakukan menggunakan pendekatan *Game Development Life Cycle*

(GDLC), mencakup analisis kebutuhan, perancangan antarmuka, pengembangan *gameplay*, dan pengacakan soal dengan algoritma *Knuth Shuffle*. Hasil implementasi menunjukkan bahwa *game* mampu menyajikan soal secara acak dan tidak repetitif, sehingga mendorong pemahaman logika alih-alih hafalan. Desain berbasis narasi, sistem penyimpanan progres, dan antarmuka interaktif menjadikan *game* ini sesuai untuk pembelajaran yang adaptif dan menarik.

Aplikasi dapat dijalankan langsung di berbagai perangkat tanpa instalasi, dengan fitur level, skor, dan misi yang memperkaya pengalaman belajar. Namun, penelitian ini masih terbatas pada tahap prototipe tanpa pengujian kuantitatif terhadap efektivitas pembelajaran. Ke depan, disarankan dilakukan evaluasi eksperimental dan penambahan fitur personalisasi agar *game* lebih adaptif terhadap kemampuan pengguna. *EduPython Adventure* diharapkan menjadi kontribusi awal dalam pengembangan media pembelajaran digital yang inovatif dan aplikatif di lingkungan SMK.

## DAFTAR PUSTAKA

- [1] M. Videnovik, T. Vold, L. Kjøning, A. Madevska Bogdanova, dan V. Trajkovik, "Game-based learning in computer science education: a scoping literature review," 1 Desember 2023, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1186/s40594-023-00447-2.
- [2] D. Rais dan Z. Xuezhai, "Elevating student engagement and academic performance: A quantitative analysis of *Python* programming integration in the Merdeka Belajar curriculum," *Journal on Mathematics Education*, vol. 15, no. 2, hlm. 495–516, 2024, doi: 10.22342/jme.v15i2.pp495-516.
- [3] H. C. Ling, K. L. Hsiao, dan W. C. Hsu, "Can Students' Computer Programming Learning Motivation and Effectiveness Be Enhanced by Learning *Python* Language? A Multi-Group Analysis," *Front Psychol*, vol. 11, Jan 2021, doi: 10.3389/fpsyg.2020.600814.
- [4] U. Saokani, M. Irfan, D. S. Maylawati, R. J. Abidin, dan I. Taufik, "Comparison of the Fisher-Yates Shuffle and the Linear Congruent Algorithm for Randomizing Questions in Nahwu Learning Multimedia," *Khazanah Journal of Religion and Technology*, vol. 1, no. 1, hlm. 10–14, 2023, doi: 10.15575/kjrt.v1i1.159.
- [5] A. Thariq dan A. Pattimura, "Penerapan Metode Fisher Yates Shuffle Pada *Game* Edukasi Pendidikan Agama Islam," *Bulletin of Computer Science Research*, vol. 4, no. 3, hlm. 290–297, 2024, doi: 10.47065/bulletincsr.v4i3.344.
- [6] C. Kirana, B. Wijaya, dan A. Holil, "Implementation of the Fisher-Yates Shuffle Algorithm in Exam-Problem Randomization on M-Learning Applications," *Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika*, vol. 7, no. 2, hlm. 47–51, 2021, doi: 10.23917/khif.v7i2.11761.
- [7] D. Hooshyar, M. Pedaste, dan Y. Yang, "Mining educational data to predict students' performance through procrastination behavior," *Entropy*, vol. 22, no. 1, hlm. 12, Jan 2020, doi: 10.3390/e22010012.
- [8] W. aulia Rohmah, A. Asriyanik, dan W. Apriyandari, "Implementation of the Algorithm Fisher Yates Shuffle on *Game* Quiz Environment," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 4, no. 1, hlm. 161–172, Jul 2020, doi: 10.31289/jite.v4i1.3863.
- [9] J. L. Plass, B. D. Homer, dan C. K. Kinzer, "Foundations of *Game*-Based Learning," *Educ Psychol*, vol. 50, no. 4, hlm. 258–283, Okt 2015, doi: 10.1080/00461520.2015.1122533.
- [10] R. C. Clark dan R. E. Mayer, *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & sons, 2023. doi: 10.1002/9781119239086.