

## BAB II TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Penelitian terdahulu ini memberikan gambaran tentang penelitian yang akan dilakukan dari perspektif penelitian terkait yang telah dilakukan sebelumnya. Dengan meninjau penelitian sebelumnya, dapat digunakan sebagai perbandingan untuk mengetahui pembaruan penelitian berikutnya dan mengusulkan beberapa terkait dengan penelitian yang akan dilakukan:

Penelitian yang dilakukan (Asriyanik & Apriyandari, 2020) dengan berjudul “Implementation of the Algorithm *Fisher Yates Shuffle* on *Games Quiz Environment*”. Penelitian ini menganalisis desain game trivia lingkungan menggunakan metode *Multimedia Development Life Cycle* (MDLC) dan pertanyaan acak algoritma *fisher-yates shuffle*. Games lingkungan ini berisi mata kuliah IPA tentang sistem organisasi kehidupan, interaksi organisme dengan lingkungan, dan pencemaran lingkungan. Komponen dari game ini adalah game material dan trivia, puzzle, dan drag and drop. Dalam penelitian ini, setiap kuesioner memiliki 10 pertanyaan acak. Hasil penerapan algoritma pengacakan *fisher-yates* pada bagian tes ini adalah soal tes bisa acak, sehingga soal/soal berbeda dan pengguna tidak mudah menebak soal berikutnya.

Penelitian yang dilakukan (Krisdiawan & Ramdhany, 2018) dengan berjudul Penerapan “Algoritma *Fisher-Yates* Pada *Games* Edukasi Pengenalan Hewan Berbasis Mobile”. Penelitian ini menggunakan algoritma *fisher-yates*. Merupakan algoritma yang menggunakan fungsi acak, sehingga nilai yang dihasilkan oleh algoritma *fisher-yates* akan memberikan hasil yang bias (Hasil pengacakan *fisher-yates* sulit untuk memprediksi mode keluaran). Pengembangan game ini mengadopsi metode pengembangan sistem RUP (*Rational Unified Process*). Dengan merancang sistem berbasis objek UML (*Unified Modeling Language*), bahasa pemrograman yang digunakan berbasis bahasa pemrograman perangkat mobile Android.

Penelitian yang dilakukan (Subaeki & Ardiansyah, 2017) dengan berjudul “Implementasi algoritma *fisher-yates shuffle* pada aplikasi multimedia interaktif untuk pembelajaran tense bahasa inggris”. Penelitian ini memberikan interaktivitas antara pengguna dan aplikasi. Interaksi dalam aplikasi pembelajaran tense ini dilakukan dalam bentuk soal-soal tiruan untuk mengasah tingkat pemahaman pengguna terhadap materi tense. Gunakan format penyimpanan data standar berbasis .XML (*Extensible Markup Language*) untuk membagi soal Tes Pemahaman menjadi dua (2) kategori, yaitu Soal Latihan dan Soal Penilaian. Algoritma acak (*shuffle*) digunakan untuk mengacak urutan pemahaman soal tes saat melakukan aplikasi. Algoritma random (pengocokan) yang digunakan adalah algoritma *fisher-yates shuffle*.

Penelitian yang dilakukan (Mulyani, 2016) dengan berjudul “Perancangan e-learning sebagai media pembelajaran pada SMP Kartika XII-1 Magelang”. Penelitian ini dilakukan dalam bentuk e-learning di SMP Kartika XIII Magelang pembelajaran bahasa Indonesia untuk mempermudah penilaian pembelajaran. Waktu pertemuan terbatas, sehingga membutuhkan layanan media online. Aplikasi ini dibuat dengan menggunakan model waterfall dan dibangun menggunakan database PHP dan MYSQL. Pada penelitian ini dilakukan pengujian aplikasi untuk mengetahui fungsionalitas aplikasi berdasarkan skenario. Namun, tanpa umpan balik pengguna, saya tidak tahu seberapa berguna aplikasi ini.

Penelitian yang dilakukan (Irawan et al., 2015) dengan berjudul “Analisa dan perancangan sistem pembelajaran online (E-learning) pada SMK Mambaul Falah Kudus”. Studi ini mengulas pemodelan *use case* diagram dan menggunakan database PHP dan MYSQL untuk membangun. Penelitian ini tidak melakukan pengujian penyebaran, jadi saya tidak tahu sejauh mana fungsi aplikasi berjalan berdasarkan skenario, karena terbatas pada desain. Selain itu, penelitian ini dilakukan tanpa umpan balik pengguna, sehingga tidak diketahui seberapa bermanfaat aplikasi tersebut.

Dari beberapa penelitian terdahulu diatas maka penulis mengambil referensi bagaimana implementasi algoritma fisher-yates pada aplikasi penilaian semester untuk pembelajaran bahasa indonesia kelas VI. Penelitian ini mengulas pembuatan Aplikasi Penilaian Semester di SDN 1 Brotonegaran Ponorogo. Aplikasi dirancang menggunakan model use case diagram dan dibangun menggunakan database PHP dan MYSQL. Penelitian ini tidak melakukan pengujian penyebaran, sehingga tidak diketahui sejauh mana fungsi aplikasi berjalan berdasarkan skenario, karena hanya dibatasi oleh desain. Selain itu, penelitian ini tidak memiliki umpan balik pengguna, jadi saya tidak tahu seberapa berguna aplikasi ini.

Pada saat yang sama, penelitian yang akan dilakukan akan mensimulasikan banyak pengguna untuk mengetahui kinerja aplikasi algoritma *fisher-yates shuffel* dalam mengelola masalah formatif siswa. Aplikasi ini dibuat dengan menggunakan model cascading dan dibangun menggunakan database PHP dan MYSQL. Pada penelitian ini dilakukan pengujian aplikasi untuk mengetahui fungsionalitas dari aplikasi berbasis skenario. Namun, tanpa umpan balik pengguna, tidak tahu seberapa berguna aplikasi ini.

## 2.2. Perancangan Sistem (UML)

Perancangan sistem adalah sistem untuk menyusun dan mengembangkan sistem informasi baru. (Yanuar Susanto, 2004). Pada tahap ini, memastikan bahwa semua prasyarat untuk menghasilkan sistem informasi baru terpenuhi. Pastikan bahwa desain memenuhi kebutuhan pengguna, dan yang paling penting, sistem perlu dikembangkan kembali. *Unified Modeling Language (UML)* adalah bahasa berbasis gambar yang digunakan untuk memvisualisasikan, mendefinisikan, membangun, dan mendokumentasikan sistem pengembangan perangkat lunak berbasis objek. *Unified Modeling Language (UML)* bukan bahasa pemrograman, tetapi model yang dibuat berhubungan langsung dengan berbagai bahasa pemrograman, sehingga dapat langsung dipetakan (*mapped*) dari model

yang dibuat menggunakan *Unified Modeling Language* (UML) yang berorientasi pemrograman obyek bahasa, seperti Java.

### 2.3. *Flowchart*

*Flowchart* adalah representasi grafis dari langkah-langkah dan urutan proses dalam suatu program. (Febriani, 2015). *Flowchart* adalah proses dan logika yang dibuat secara sistematis dari gambaran grafis dari proses pengolahan informasi atau langkah-langkah dan urutan program. *Flowchart* membantu analis dan programmer untuk memecahkan masalah yang lebih kecil dan membantu analisis pilihan lain. (Murdock, 2018).

*Flowchart* Sistem adalah urutan proses dalam sistem, yang mewakili alat-alat dari media input dan output dan jenis media penyimpanan dalam proses pengolahan data, dan diagram alir program adalah diagram dengan simbol-simbol tertentu, menggambarkan urutan proses. dari a Hubungan antara proses (instruksi) dan proses lainnya dalam program.










Jika seseorang analis dan programmer yang akan membuat *flowchart*, terdapat beberapa petunjuk yang harus diperhatikan, seperti berikut (Febriani, 2015):

1. Diagram alir dibuat menurut alur dari atas ke bawah dan kiri ke kanan.
2. Aktivitas yang dijelaskan harus didefinisikan secara hati-hati dan definisi tersebut harus dapat dipahami oleh pembaca.
3. Setiap kegiatan yang dimulai dan diakhiri harus didefinisikan dengan jelas.
4. Setiap langkah kegiatan harus dijelaskan dengan deskripsi kata kerja.
5. Langkah-langkah dalam setiap kegiatan harus dilakukan dengan urutan yang benar.
6. Ruang lingkup dan ruang lingkup kegiatan yang dijelaskan harus dieksplorasi dengan hati-hati. Cabang-cabang yang memotong kegiatan yang dijelaskan tidak perlu digambar pada diagram alir yang sama. Jika cabang tidak ada hubungannya dengan sistem, Anda harus menggunakan

prompt konektor dan meletakkan cabang pada halaman terpisah atau melewatkannya sama sekali.

7. Gunakan simbol diagram alir standar.

**Tabel 2.1** Simbol *Flowchart*

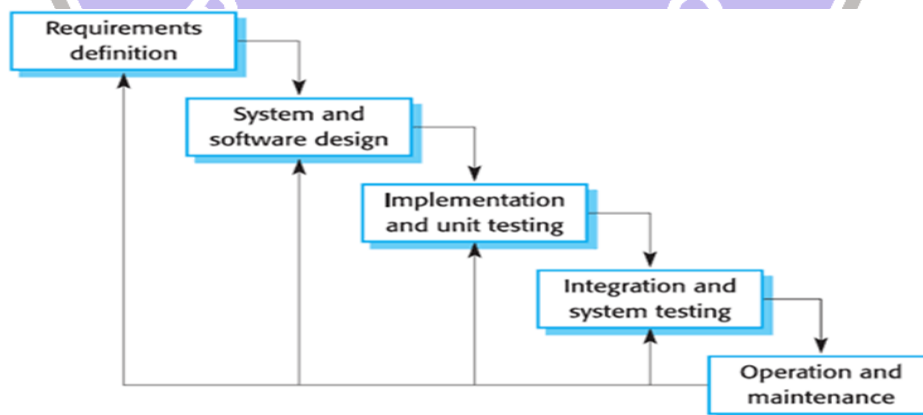
Simbol	Nama	Fungsi
	Terminator	Memulai dan mengakhiri program
	Garis Alir ( <i>Flow Line</i> )	Digunakan untuk menentukan arah aliran program.
	<i>Preparation</i>	Digunakan untuk memproses harga awal
	<i>Process</i>	Untuk proses perhitungan / proses pengolahan data
	<i>Input/Output Data</i>	Digunakan untuk mengolah data input/output, parameter dan informasi.
	<i>Predefined Process</i> (Sub Program)	Jadikan subrutin/proses dijalankan dari subrutin
	<i>Decision</i>	Untuk membandingkan pernyataan, pilih data yang menyediakan opsi pemrosesan lebih lanjut.
	<i>On Page Connector</i>	Digunakan untuk menghubungkan bagian flowchart pada satu halaman.
	<i>Off Page Connector</i>	Digunakan untuk menghubungkan bagian-bagian dari flowchart pada halaman yang berbeda.

Sumber: (Murdock, 2018)

## 2.4. Waterfall

Metode *waterfall* adalah metode yang merekomendasikan pendekatan sistematis dan berurutan untuk membangun perangkat lunak melalui berbagai tahapan SDLC. Pengembangan sistem secara keseluruhan dilakukan melalui beberapa tahapan/tahapan. Metode pengembangan perangkat lunak juga disebut siklus hidup pengembangan perangkat lunak (SDLC). Metode waterfall merupakan metode pengembangan perangkat lunak yang paling tua karena sifatnya yang natural. Metode waterfall adalah metode SDLC tertua yang digunakan dalam pengembangan perangkat lunak. Urutan dalam metode *waterfall* adalah berkesinambungan, mulai dari proses perencanaan, analisis, desain dan implementasi dalam sistem.

Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu di sebut *waterfall* (Air Terjun). (Sommerville, 2011) menjelaskan bahwa ada lima tahapan pada Metode Waterfall, yakni *Requirements Analysis and Definition*, *System and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing*, dan *Operation and Maintenance*.



Gambar 2.1 Waterfall

Berikut adalah penjelasan dari tahap – tahap yang dilakukan dalam metode waterfall:

- a. Tahap analisis, merupakan bagian awal dari pengerjaan suatu proyek perangkat lunak. Dimulai dengan mempersiapkan data-data yang diperlukan.
- b. Tahap perancangan,  
Tahap pengkodean, yaitu pada tahapan ini penulis melakukan pengkodean serta pengujian sistem. Pada tahapan pengkodean atau implementasi bahasa pemrograman yang telah dipilih sebelumnya yaitu bahasa pemrograman Java *native* serta menggunakan software Android Studio untuk membuat program yang sesungguhnya.
- c. Tahap implementasi dan pengujian, pada tahapan ini penulis melakukan menyerahkan aplikasi yang telah selesai dibuat kepada klien kemudian di lakukan instalasi serta pelatihan sebagai support penulis kepada klien. Yang selanjutnya untuk tahapan pengujian, penulis memilih metode *black box*.
- d. Pemeliharaan, yaitu penerapan sistem secara keseluruhan disertai pemeliharaan jika terjadi perubahan struktur, baik dari segi *software* maupun *hardware*.

## 2.5. Algoritma Fisher-Yates

*Fisher-Yates Yhuffle* (diambil dari nama Ronald Fisher dan Frank Yates) atau Knuth shuffle (diambil dari nama Donald Knuth), adalah algoritma yang menghasilkan susunan acak dari himpunan berhingga, dengan kata lain mengacak himpunan tersebut. Jika *fisher-yates shuffle* diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan satu sisi, sehingga setiap permutasi memiliki probabilitas yang sama. (Ade-Ibijola, 2012).

*Fisher-Yates Shuffle* atau atau cara pengacakan populasi hingga. Implementasi *Fisher-Yates* dapat digunakan dua bentuk, yaitu: metode asli dan metode modern. Metode asli diterapkan secara berulang ke item daftar

input, dan kemudian menulis ulang daftar output kedua (pavel, Mikcka 2011).

Algoritma *Fisher-Yates* diseleksi sebab ialah tata cara pengacakan yang lebih baik, ataupun dapat dikatakan sesuai buat pengacakan angka, waktu eksekusi kilat, serta pengacakan tidak memerlukan banyak waktu. Algoritma ini memiliki dua metode yang satu asli dan modern. Namun, selama aplikasi ini dikembangkan, algoritma diimplementasikan kedalam metode yang modern. Metode ini menggunakan sistem pengacakan computer sehingga dipilih, karena hasil pengacakan bisa lebih beragam.

Algoritma modern berbeda dengan algoritma lama. Digit terakhir dari algoritma lama akan ditransfer ke nomor lotre, dan nomor lotre akan diubah untuk dihapus setiap kali lotre ditarik dan dilanjutkan ke iterasi berikutnya.

Metode *Fisher-Yates Shuffle* secara umum adalah sebagai berikut :

1. Inisialisasi jumlah elemen dalam array.
2. Ambil elemen secara acak dari elemen yang tersisa
3. Kemudian tukar dengan elemen saat ini.

Metode acak dari *Fisher-Yates* adalah metode untuk membuat permutasi acak pada beberapa keuanggan, dengan kata lain untuk mengenkripsi sesuatu dari koleksi.

Misalkan dalam game kartu bridge terdapat 10 kartu yang diacak, hingga array-nya adalah urutan Kartu = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]. Array tersebut dimasukkan kedalam prosedur pengacakan dimana proses pengacakan terjadi, kemudian didapat panjang array yang setelah itu dimasukkan ke dalam variabel  $m$  ( $m=10$ ).

Langkah-langkah metode metode pengacakan *Fisher-Yates* untuk mengacak *puzzle* adalah sebagai berikut :

1. Ambil satu elemen secara acak dari elemen yang tersisa. Pengambilan elemen acak adalah berdasarkan elemen yang tersisa. Misalkan jika  $m=10$ , maka elemen acak yang boleh diambil adalah 10 ( $\text{array}[0\dots9]$ ).
2. Tukar dengan elemen saat ini. Penukaran dilakukan dengan memasukkan elemen saat ini ke dalam suatu variabel baru bernama  $t$ .



(*array*[m]) diisi nilai dari elemen acak tadi (*array*[i]) dan elemen acak (*array*[i]) diisi nilai dari variabel t.

3. Ulangi selama masih ada elemen yang tersisa (Yuningsih et al., 2018)

