

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Berikut ini adalah beberapa penelitian yang dilakukan oleh peneliti-peneliti terdahulu. Penelitian berikut adalah penelitian yang berkaitan mengenai yang penulis buat :

Penelitian dari Beki Subaeki dan Dicky Ardiansyah (2017) yang berjudul ”Implementasi Algoritma *Fisher-Yates Shuffle* pada Aplikasi Multimedia Interaktif untuk Pembelajaran *Tenses* Bahasa Inggris”. Dalam penelitian tersebut metode *Fisher-Yates Shuffle* diterapkan untuk mengacak urutan soal uji pemahaman *tenses* dalam bentuk PBK (Pembelajaran Berbantuan Komputer) yang berbasis Multimedia Interaktif. Soal uji pemahaman dibuat dalam dua kategori, yakni soal latihan dan soal tes evaluasi menggunakan format penyimpanan data standar berbasis .XML (*eXtensible Markup Language*).

Penelitian dari Akbar Gani dan Linda Marlinda (2017) yang berjudul ”Aplikasi Pembelajaran Trigonometri berbasis Android Menggunakan Algoritma Fisher Yates Shuffle”. Dalam penelitian tersebut metode *Fisher Yates Shuffle* diterapkan untuk mengacak urutan soal uji pemahaman pembelajaran Trigonometri. Dalam penelitian tersebut aplikasi yang dibangun berbasis android dengan menggunakan permodelan UML. Aplikasi ini berjalan di OS Android *Jelly Bean* sampai versi yang terbaru.

Penelitian dari Yoaki, dkk (2018) yang berjudul ”Implementasi Algoritma *Fisher-Yates Shuffle* pada Aplikasi Media Interaktif Pengenalan Nama-nama Negara di Benua Asia Berbasis Multimedia”. Dalam penelitian tersebut metode *Fisher-Yates Shuffle* diterapkan untuk mengacak latihan soal tentang nama-nama negara di benua Asia. Metode pengumpulan data yang digunakan dalam penelitian tersebut yaitu dengan cara observasi, di mana pengamatan dilakukan secara langsung melalui wawancara. Dalam prancangan aplikasi menggunakan *Flowchart* dan *Storyboard*.

II.2. Landasan Teori

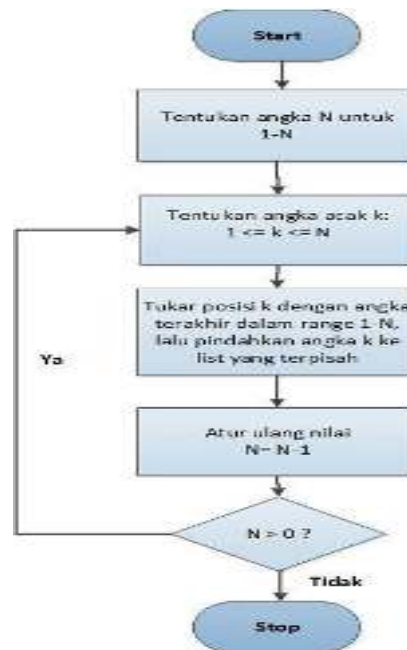
II.2.1. Algoritma *Fisher Yates Shuffle*

Algoritma *Fisher Yates Shuffle* adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan berhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Jika algoritma *Fisher Yates Shuffle* diimplementasikan dengan benar maka akan menghasilkan algoritma yang tidak berat sebelah, sehingga setiap permutasi memiliki kemungkinan yang sama.

Menurut Loman dan Wiradinata (2014:52) “Algoritma *Fisher-Yates Shuffle* adalah algoritma yang digunakan untuk mengacak urutan sekelompok angka”. Tahapan dari algoritma ini adalah:

1. Menuliskan angka dari 1 sampai N Memilih satu angka k secara acak dan menuliskan di urutan angka baru
2. Mencoret k dari urutan angka yang sebelumnya
3. Mengulang tahapan kedua sampai angka habis (Akbar Gani dan Linda Marlinda : 2017)

Dalam proses pengacakan, Algoritma *Fisher Yates Shuffle* akan dijelaskan menggunakan *Flowhart*. Akan dijelaskan pada Gambar II.I



Gambar II.I. Flowchart Algoritma Fisher Yates Shuffle

Dari gambar II.I, diketahui bahwa algoritma *Fisher Yates Shuffle* dalam mengacak urutan data diawali dengan menentukan angka N untuk range $1-N$. Setelah itu tentukan angka acak k dari range $1-N$, tidak boleh 0 dan melebihi nilai N . Tukar posisi k dengan angka terakhir range $1-N$ ditempat terpisah. Atur ulang N , $N=N-1$. Jika N bernilai 0 , maka hentikan proses pengacakan. Algoritma ini diaplikasikan dan disempurnakan pada sistem komputer oleh Richard Durstenfeld pada tahun 1964 Pada metode baru ini, angka tidak dicoret dari urutan angka lama, melainkan menempatkan angka terakhir di tempat angka yang telah dipindah ke urutan angka baru.

Dalam pengembangan perangkat lunak ini, penulis menggunakan metode modern dari algoritma *Fisher Yates Shuffle*. Pada modern *method* dijabarkan untuk penggunaan komputerisasi yang dikenalkan oleh Richard Durstenfield pada tahun 1964. Modern *method* lebih optimal, sangat komputasi dan matematis, atau bisa dikatakan bahwa metode ini khusus digunakan untuk pengacakan dengan sistem komputerisasi sehingga pengacakan bisa lebih variatif. Dari dua metode tersebut, yang penulis gunakan adalah metode modern dikarenakan bahwa metode ini memang dikhususkan untuk pengacakan dengan sistem komputerisasi agar hasil pengacakan bisa lebih variatif.

Berikut ini adalah pengerjaan dari versi modern. *Range* adalah jumlah angka yang belum terpilih, *roll* adalah angka acak yang terpilih, *scratch* adalah daftar angka yang belum terpilih, *result* adalah hasil permutasi yang akan didapatkan.

Tabel II.1 Algoritma *Shuffle Fisher Yates*

Range	Roll	Scratch	Result
		1 2 3 4 5	
1 – 5	3	1 2 5 4	3
1 – 4	4	1 2 5	4 3
1 – 3	2	1 5	2 4 3
1 – 2	1	5	1 2 4 3

Dari tabel diatas, maka didapatkan hasil permutasi **5 1 2 4 3**. Sehingga walaupun soal sama, namun urutan soal akan ditampilkan berbeda oleh program pada halaman Soal Uji Kompetensi.

II.2.2. Microsoft Excel

Microsoft Excel adalah salah satu produk perangkat lunak pengolah data dari Microsoft dalam bentuk lembaran tabel yang tersebar (*spreadsheet*). Perangkat lunak ini merupakan bagian dari paket perangkat lunak Microsoft Office yang di dalamnya terdapat banyak perangkat lunak (keperluan perkantoran) misalnya: *Microsoft Word* (pengolah kata), *Microsoft Power Point* (untuk keperluan presentasi), dan *Microsoft Access* (untuk keperluan basis data).

Microsoft Excel digunakan untuk mengolah berbagai jenis atau tipe data misalnya data nilai pelajaran siswa yang berarti jenis datanya numerik, data teman dan nomor telepon yang berarti jenis datanya teks, data rekapitulasi pemakaian barang dalam bentuk angka dan teks, dll. Dalam Excel selain bisa digunakan untuk mencatat dan menyimpan data-data tersebut, Excel juga mampu melakukan perhitungan penjumlahan, perkalian, fungsi-fungsi logika, perhitungan rata-rata, bahkan sampai pembuatan grafik.

(<https://staffnew.uny.ac.id/upload/132161227/pengabdian/materi-ppmmicrosoft-excell.pdf>)

II.2.3. Formula

Formula adalah satu set perintah untuk menghasilkan *output* tertentu. Di dalam perintah itu, mungkin kita memasukkan berbagai macam fungsi untuk menghasilkan *output* tertentu. Sebagai contoh, jika kita ingin mengalihkan penjumlahan di sel A1 hingga A10 dan B1 hingga B10, kita bisa tulis formula: **=SUM(A1:A10)*SUM(B1:B10)**. *Formula* juga bisa ditulis tanpa melibatkan fungsi sama sekali. Sebagai contoh, set perintah berikut ini juga merupakan

sebuah formula walaupun tidak melibatkan fungsi: $=A1+A2$. (Jubilee Enterprise : 2019).

II.2.4. Aplikasi

Menurut Hasan Abdurahman dan Asep Ririh Riswaya (2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. (Adi Widarma dan Sri Rahayu : 2017)

II.2.5. Aplikasi *Mobile*

Menurut *Mobile Marketing Association* (2015) Aplikasi *Mobile* adalah perangkat lunak yang berjalan pada perangkat *mobile* seperti *smartphone* atau *tablet PC*. Aplikasi *Mobile* juga dikenal sebagai aplikasi yang dapat diunduh dan memiliki fungsi tertentu sehingga menambah fungsionalitas dari perangkat *mobile* itu sendiri. Untuk mendapatkan *mobile application* yang diinginkan, *user* dapat mengunduhnya melalui situs tertentu sesuai dengan sistem operasi yang dimiliki. *Google Play* dan *iTunes* merupakan beberapa contoh dari situs yang menyediakan beragam aplikasi bagi pengguna Android dan iOS untuk mengunduh aplikasi yang diinginkan. (<https://jurnaluntan.ac.id/index.php/justin/article/download/9984/9752>)

II.2.6. *Android*

Menurut Safaat H (2014:1) Android adalah sebuah operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android sebagai “*platform mobile* pertama yang lengkap, terbuka dan bebas”.

1. Lengkap (*Complete Platform*) para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform*) *Platform* Android disediakan melalui open source. Pengembangan dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux kernel 2.6
3. *Free (Free Platform)* Android adalah *platform* aplikasi yang bebas untuk *develope*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada kontrak yang diperlukan. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

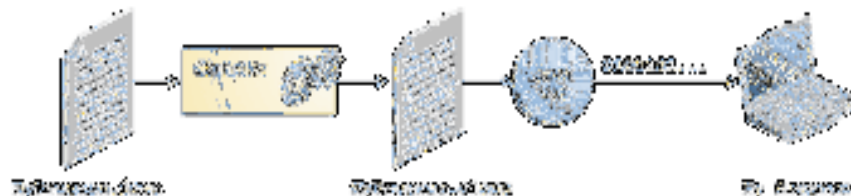
Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkan. Pengembang aplikasi Android bebas mendistribusikan aplikasi mereka dibawah skema lisensi yang mereka inginkan. (Akbar Gani dan Linda Marlinda : 2017)

II.2.7. Java

Teknologi Java adalah bahasa pemrograman dan juga platform. Bahasa pemrograman Java adalah bahasa tingkat tinggi yang dapat dicirikan dengan semua kata kunci berikut:

1. Simple
2. Object oriented
3. Distributed
4. Multithreaded
5. Dynamic
6. Architecture neutral
7. Portable
8. High performance
9. Robust
10. Secure

Setiap kata kunci sebelumnya dijelaskan dalam *The Java Language Environment*, sebuah buku putih yang ditulis oleh James Gosling dan Henry McGilton. Dalam bahasa pemrograman *Java*, semua kode sumber pertama kali ditulis dalam file teks biasa yang diakhiri dengan ekstensi *.java*. File sumber tersebut kemudian dikompilasi ke dalam file *.class* oleh kompiler *javac*. File *.class* tidak mengandung kode yang asli untuk prosesor anda melainkan berisi *bytecodes* - bahasa mesin *Java Virtual Machine* (Java VM). Alat peluncur *java* kemudian menjalankan aplikasi Anda dengan mesin *Java Virtual Machine*.



Gambar II.2 Tinjauan umum proses pengembangan perangkat lunak.

(Sumber: [https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.ht](https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html)

ml)

II.2.8. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA* . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

1. Sistem versi berbasis *Gradle* yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
5. *Template* kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.

II.2.9. *Unified Modeling Language (UML)*

Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem, dikutip oleh (Ade Hendini, 2016).


Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

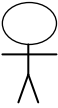


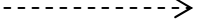
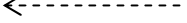
1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Simbol-simbol yang digunakan dalam *use case diagram*, yaitu :

Tabel II.2. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>




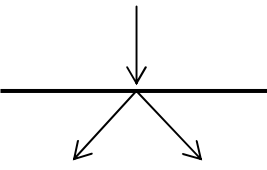
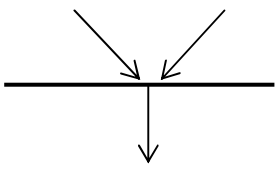
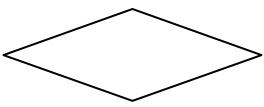

	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Ade Hendini : 2016 : 108)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.3. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini : 2016 : 109)