

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terkait

Pada penelitian terkait yang dilakukan oleh Arya Jaya Alamsyah, dkk (2021) dengan judul penelitian Prototipe Sistem Computer Based Test dengan Pengacakan Soal Menggunakan Metode Fisher-Yates Shuffle. Laporan perancangan aset karakter pada game “Apostle” Pengembangan Sistem CBT dengan pengacakan soal dibuat dengan berbasis web dan menggunakan FYS. Hasil percobaan yang dilakukan adalah dengan mengacak sebuah soal USBN mata pelajaran Matematika berjumlah 30 soal yang diacak sebanyak 40 kali dan menghasilkan urutan soal yang berbeda. Hasil pengujian sistem CBT menjelaskan sistem CBT yang dibuat sudah 100% berjalan sesuai dengan fungsionalitasnya masing-masing. Adapun yang membedakan pada penelitian yang akan dibangun adalah aplikasi dibangun nantinya dapat diterapkan pada smartphone android dengan jumlah soal berdasarkan materi dengan jumlah setiap jumlah soal minimal 25 soal.

Pada penelitian terkait yang dilakukan oleh Fransiskus Panca Juniawan & Hengki (2019) dengan judul penelitian PENGACAKAN SOAL UJIAN PENERIMAAN POLRI MENGGUNAKAN ALGORITME FISHER YATES SHUFFLE. Sistem yang digunakan adalah Ujian Masuk berbasis Komputer (*Computer Base Test*). Adapun yang membedakan pada penelitian yang akan dibangun adalah aplikasi ujian yang dapat mengacak soal-soal ujian dengan

menggunakan metode Fisher Yates berbasis android, yang dimana dalam penggunaan dapat diterapkan secara online pada *smartphone android*.

Penelitian terkait yang dilakukan oleh Rio Priantama & Yuda Priandani (2019) dengan judul IMPLEMENTASI ALGORITMA FISHER YATES UNTUK PENGACAKAN SOAL PADA APLIKASI MOBILE LEARNING KUIS FIQIH BERBASIS ANDROID. Aplikasi mobile learning kuis fiqih dapat membantu proses pembelajaran ilmu fiqih. Pengacakan menggunakan algoritma Fisher Yates dapat diterapkan di dalam aplikasi mobile learning kuis fiqih sebagai pengacak soal yang akan muncul dalam setiap keluarnya soal tidak berulang. Adapun yang membedakan pada penelitian yang akan dibangun adalah suatu aplikasi yang dikhususkan untuk ujian online pada sekolah berbasis android. Dimana aplikasi nanti nya hanya menampilkan soal-soal ujian.

Penelitian terkait yang dilakukan oleh Rendy Rian Chrisna Putra & Tri Sugihartono (2019) dengan judul penelitian PENERAPAN ALGORITMA FISHER-YATES SHUFFLE PADA COMPUTER BASED TEST UJIAN SEKOLAH DI SMKN 1 PAYUNG. Aplikasi Test Berbasis Komputer di SMKN 1 Payung sudah berhasil dibuat oleh penulis yang berisi mengenai pengacakan soal, sehingga dengan adanya algoritma fisher yates shuffle ini dapat membantu meminimalisir adanya kecurangan menyontek yang biasa dilakukan oleh siswa. Adapun yang membedakan pada penelitian yang akan dibangun adalah aplikasi ujian yang dapat mengacak soal-soal ujian dengan menggunakan metode Fisher Yates berbasis android, yang dimana dalam penggunaan dapat diterapkan secara online pada *smartphone android*.

Penelitian terkait yang dilakukan oleh Mhd Arief Hasan, dkk (2017) dengan judul Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru (Studi Kasus : Universitas Lancang Kuning Riau). Penerapan algoritma Fisher-Yates yang digunakan pada aplikasi CBT (Computer Based Testing) dapat mengacak soal yang terlihat pada perbedaan tampilan soal pada setiap peserta ujian sehingga dalam pelaksanaan ujian setiap mahasiswa dalam menjawab soal memiliki nomor yang sama tetapi bentuk soal yang berbeda. Adapun yang membedakan pada penelitian yang akan dibangun adalah aplikasi ujian yang dapat mengacak soal-soal ujian dengan menggunakan metode Fisher Yates berbasis android, yang dimana dalam penggunaan dapat diterapkan secara online pada smartphone android.

II.2 Landasan Teori

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan.

II.2.1 Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu melayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018).

II.2.2 Ujian

Ujian merupakan salah satu syarat siswa dalam menempuh jenjang pendidikan supaya bisa naik menuju tingkat pendidikan di atasnya, setelah melakukan ujian pastinya para siswa menunggu hasil ujian yang telah mereka lakukan. Bahkan para orang tua harus menunggu cukup lama untuk mendapatkan hasil ujian sebelum dijadikan laporan kenaikan kelas. Selain itu, orang tua juga tidak bisa mendapatkan informasi tentang perkembangan anaknya di sekolah. Sehingga hasil belajar anak tersebut tidak bisa didapatkan secara maksimal karena kurangnya pendampingan orang tua untuk memperbaiki hasil belajar anaknya (Ahmad Shodikun & Elgamar Syam; 2018).

II.2.3 Software

Software merupakan komponen non fisik komputer berupa kumpulan program dan struktur datanya. Software dibagi menjadi dua kategori, yaitu sistem program dan aplikasi. Language Program (bahasa program) Bahasa program merupakan software yang dipergunakan untuk memprogram atau memberi instruksi kepada komputer agar menjalankan perintah yang dimasukkan (Korni Mufarola; 2019).

II.2.4 Android

Android adalah aplikasi sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk

menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak (Deny Adhar; 2019).

II.2.4.1 Versi Android

1. *Android 1.5 Cupcake*

Cupcake dirilis 30 April 2009. *Cupcake* menjadi versi android pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun Google memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5 *Cupcake* adalah kue kecil yang dipanggang dalam cetakan berbentuk cup.

2. *Android 1.6 Donut*

Android V1.6, codename Donut, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan reboot, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin. Bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan toping di atasnya.

3. *Android 2.0/2.1 Eclair*

Android 2.0/2.1 Eclair Dirilis 26 Oktober 2009. *Eclair* adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya.

4. *Android 2.2 Froyo*

Dirilis 20 Mei 2010. Menggunakan codename *Froyo*, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari Yoghurt.

Froyo singkatan dari Frozen Yoghurt, Froyo adalah yoghurt yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim.

5. *Android 2.3 Gingerbread*

Android versi 2.3 Gingerbread dirilis resmi tanggal 6 Desember 2010. Gingerbread merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. *Android 3.0 Honeycomb*

Dirilis tanggal 22 February 2011. H adalah sereal sarapan manis yang sudah dibuat oleh Posting Sereal. Seperti namanya, Honeycomb/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu.

7. *Android 4.0 Ice Cream Sandwich*

Android 4.0-4.0.2 API Level 14 dan 4.0.3 API Level 15 pertama dirilis 19 Oktober 2011. Dinamai Ice Cream Sandwich. Ice Cream Sandwich es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.

8. *Android 4.1 Jelly bean*

Android Jelly Bean diluncurkan pertama kali pada Juli 2012, dengan berbasis Linux Kernel dari Android 4.1 API Level 16, Android 4.2 API Level 17, Android 4.3 API Level 18. Penamaan mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

9. *Android 4.4 KitKat*

Android 4.4 Kitkat API level 19. Google mengumumkan *Android KitKat* (dinamai dengan izin Nestle dan Hershey) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. *KitKat* merupakan merk sebuah coklat yang dikeluarkan oleh Nestle. Rilis berikutnya setelah nama *KitKat* diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai 'Pie' (Hendra Nugraha Lengkong; 2016).

II.2.5 *Android Studio*

Android Studio adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Aplikasi *Android* yang dikembangkan menggunakan *Java* dan mudah menyesuaikan ke platform baru. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, dan fungsi-fungsi pustaka, hingga perangkat lunak API (*Application Programming Interface*), termasuk aplikasi SDK (*Software Development Kit*). *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan dan diadaptasikan untuk mendukung berbagai konfigurasi perangkat keras. Sistem operasi yang mendasari *Android* merupakan lisensi di bawah naungan GNU, *General Public License Versi 2 (GPLv2)*, yang biasa dikenal dengan istilah *Copy left*. Istilah *copy left* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh di bawah terms. Distribusi *Android* berada di bawah lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua atau seterusnya. Pengembang aplikasi *Android*

diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. (Ganda Yoga Swara, dkk; 2017) .

II.2.5.1 *Integrated Development Environment (IDE)*

Integrated Development Environment adalah aplikasi pengembang perangkat lunak dengan fungsi-fungsi terintegrasi yang dibutuhkan untuk membangun sebuah perangkat lunak seperti code editor, debugger, compiler, dan sebagainya (Efmi Maiyana; 2018).

II.2.5.2 *Android Software Development Kit (Android SDK)*

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk memulai pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Pada Android SDK ini terdiri dari debugger, libraries, handset emulator, dokumentasi, kode contoh dan tutorial. SDK memungkinkan pengembang membuat aplikasi untuk platform Android. SDK, Android mencakup proyek sampel dengan kode sumber, perangkat pengembangan, emulator dan perpustakaan yang diperlukan untuk membangun aplikasi Android. Aplikasi yang ditulis dengan bahasa pemrograman Java dan berjalan di Dalvik, mesin virtual yang dirancang khusus untuk penggunaan embedded yang berjalan diatas kernel Linux (Efmi Maiyana; 2018).

II.2.5.3 Java Development Kit (JDK)

JDK (*Java Development Kit*) adalah Paket fungsi API untuk bahasa pemrograman Java, meliputi *Java Runtime Environment* (JRE) dan *Java Virtual Machine* (JVM) (Efmi Maiyana; 2018).

II.2.5.4 Android Virtual Device (AVD)

Android Virtual Device merupakan emulator untuk menjalankan program aplikasi Android yang kita buat. AVD ini selanjutnya digunakan sebagai tempat untuk test dan menjalankan aplikasi Android tanpa harus menggunakan perangkat Android yang sebenarnya. Sebelum menggunakan AVD harus menentukan karakteristiknya, misalkan dalam menentukan versi Android, jenis dan ukuran layar dan besarnya memori. AVD bisa dibuat sebanyak yang kita inginkan (Efmi Maiyana; 2018).

II.2.6 Fisher Yates

Fisher Yates (diambil dari nama Ronald *Fisher* dan Frank *Yates*) atau juga dikenal dengan nama *Knuth Shuffle* (diambil dari nama Donald Knuth), adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Jika diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan berat sebelah, sehingga setiap permutasi memiliki kemungkinan yang sama.

Proses dasar dari *Fisher-Yates Shuffle* mirip dengan mengambil kartu dari setumpuk kartu satu demi satu sampai tidak ada yang tersisa. Algoritma *Fisher-*

Yates dianggap oleh banyak orang sebagai metode yang objektif dan optimal untuk menghasilkan permutasi yang benar-benar acak dari himpunan berhingga (Ihya Ulumuddin Banyumanis & Devi Fitriana; 2017).

Pada algoritma ini tidak terdapat rumus seperti Algoritma lainnya, melainkan hanya memiliki langkah-langkah untuk menyelesaikannya. Analisis Algoritma Misalkan pada level 1 ada 4 soal yang akan diacak, maka array-nya adalah urutan soal = [0, 1, 2, 3,]. Array tersebut dimasukkan kedalam prosedur shuffle dimana proses pengacakan terjadi. Dari array tersebut didapat panjang array yang kemudian dimasukkan ke dalam variabel m ($m = 10$). Langkah-langkah pengacakan soal dengan Fisher-Yates Shuffle adalah sebagai berikut :

1. Ambil satu elemen secara acak dari elemen yang tersisa. Pengambilan elemen acak adalah berdasarkan elemen yang tersisa. Misalkan jika $m = 3$, maka elemen acak yang boleh diambil adalah 3 ($\text{array}[0 \dots 2]$).
2. Tukar dengan elemen saat ini. Penukaran dilakukan dengan memasukkan elemen saat ini ($\text{array}[m]$) ke dalam suatu variabel baru bernama t . Kemudian elemen saat ini ($\text{array}[m]$) diisi nilai dari elemen acak tadi ($\text{array}[i]$). Dan elemen acak ($\text{array}[i]$) diisi nilai dari variabel t .
3. Ulangi selama masih ada elemen yang tersisa. Selanjutnya dilakukan pengulangan sebanyak 4 kali untuk mengacak urutan soal tersebut (Fauji Ahmad; 2018).

II.2.7 UML (*Unified Modeling language*)

Unified Modeling Language (UML), adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena *UML* menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

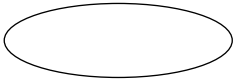
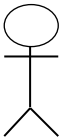
UML merupakan keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Pemodelan *visual* adalah salah satu cara berpikir tentang persoalan menggunakan model-model yang diorganisasikan seputar dunia nyata yang berguna untuk memahami persoalan, mengkomunikasikan dengan orang-orang yang terlibat dalam proyek (*costumer*, ahli dibidangnya, analisis, desainer dan lain-lain). Serta didefinisikan sebagai proses pemodelan sistem informasi menggunakan pengaturan standar elemen grafik dan objek-objek dalam sistem dan antar sistem itu sendiri. (Munawar; 2018: 49).

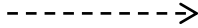
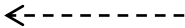
1. *Use Case* Diagram

Use Case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *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 dapat dilihat pada tabel II.1:

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<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>.
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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 <i>control</i> terhadap <i>Use Case</i>.
	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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>Use Case</i> yang

	menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<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.
	<i>Extend</i> , merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.

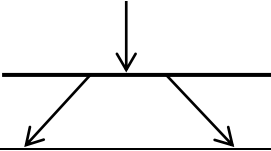
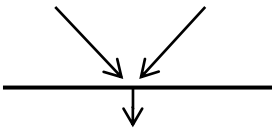
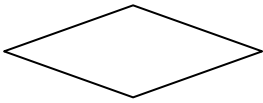
(Sumber : Munawar; 2018: 89)

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* dapat dilihat pada tabel II.2 :

Tabel II.2. 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.

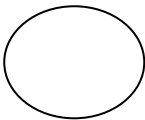
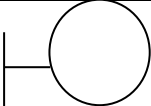
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

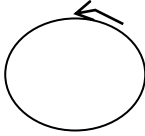
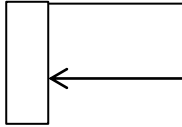
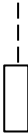
(Sumber : Munawar ; 2018 : 137)

3. Diagram Urutan (*Sequence Diagram*)

Diagram urutan menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada tabel II.3:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor

	dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Munawar ; 2018 : 186)

