
**PENERAPAN ALGORITMA FISHER-YATES PADA COMPUTER BASED TEST DI
PANDA MANDARIN COURSE**

Oleh

Yeremia Firhan Subagja¹, RZ Abdul Aziz²

^{1,2}Jurusan Teknik Informatika, Institut Informatika dan Bisnis Darmajaya

Email: yfirhan,yf@gmail.com, rz_aziz@darmajaya.ac.id

Abstrak

Panda Mandarin Course merupakan tempat kursus belajar Bahasa Mandarin yang memiliki 3 tingkatan kelas, yaitu *Elementary 1*, *Elementary 2*, dan *Elementary 3* dimana setiap kelas memiliki tingkat kesulitan yang berbeda-beda. Pada saat ini sistem ujian yang diterapkan di Panda Mandarin Course adalah ujian secara konvensional. Namun ujian secara konvensional memiliki beberapa kekurangan, yaitu guru harus membuat soal setiap kali ujian dilaksanakan, membutuhkan banyak kertas, dan guru harus memeriksa ujian terlebih dahulu untuk mengetahui hasil yang di dapat setiap murid. Untuk mengatasi hal tersebut, dibangun sebuah sistem yang dapat mempermudah jalannya ujian. Sistem ini disebut *Computer Based Test*. Dengan adanya *Computer Based Test*, maka ujian dapat dijalankan dengan komputer, sistem juga dapat menilai hasil ujian secara langsung setelah murid menyelesaikan ujian. Sistem ini juga diimplementasikan dengan algoritma *Fisher-Yates* agar setiap soal dan jawaban yang diterima oleh murid dapat teracak. Hal ini dilakukan untuk mengurangi tindakan kecurangan yang dilakukan oleh murid. Setelah sistem dianalisis, terjadi kemunculan soal yang sama sebanyak 44% dan terjadi kemunculan soal yang sama pada nomor urut yang sama sebanyak 14% yang mengindikasikan kemungkinan terjadinya kemunculan soal yang sama di waktu yang bersamaan pada 2 orang murid sangatlah kecil. Juga diketahui rata-rata kemunculan soal adalah sebesar 3,33%.

Kata Kunci: *Computer Based Test, Fisher-Yates, Prototype*

PENDAHULUAN

Panda Mandarin Course merupakan tempat kursus belajar Bahasa Mandarin. Panda Mandarin memiliki 3 tingkatan kelas, yaitu *Elementary 1*, *Elementary 2*, dan *Elementary 3*. Setiap kelas memiliki tingkat kesulitan yang berbeda-beda.

Ujian merupakan salah satu cara untuk melakukan proses evaluasi belajar. Melalui ujian seseorang juga dapat mengukur taraf pencapaian proses pembelajaran sehingga murid dan instansi dapat memahami tingkat kemampuannya dalam memahami bidang studi yang telah ditempuh [1]. Ujian dapat dilakukan dengan dua cara, yaitu ujian secara konvensional dan ujian atau test berbasis komputer (*Computer Based Test*).

Dengan memanfaatkan teknologi saat ini, ujian secara konvensional dapat diganti

dengan ujian berbasis komputer. Perbedaan dari kedua cara tersebut adalah ujian berbasis komputer tidak perlu menggunakan kertas untuk menyampaikan soal dan menulis jawaban. Namun ujian konvensional dan ujian berbasis komputer memiliki kelemahan yang sama, yaitu terdapat kemungkinan perbuatan curang yang dilakukan oleh murid, karena soal masih disampaikan secara berurutan. Untuk mengatasi hal tersebut, dibutuhkan algoritma yang dapat mengacak soal ujian.

Algoritma *Fisher-Yates* adalah sebuah algoritma yang menghasilkan permutasi acak dari satu set terbatas [2]. Hasil dari algoritma ini tidak akan berat sebelah jika diimplementasikan dengan benar, sehingga setiap permutasi memiliki kemungkinan yang sama [3]. Dengan menggunakan algoritma *Fisher-Yates* pada test berbasis komputer

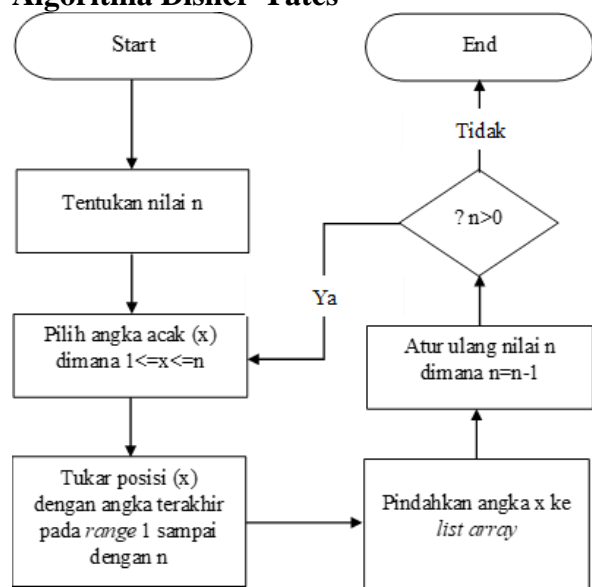
(Computer Based Test) maka soal-soal yang diberikan akan teracak urutannya. Berdasarkan masalah tersebut penulis akan mencoba membangun sistem ujian berbasis komputer yang diimplementasikan dengan algoritma Fisher-Yates untuk mengacak soal ujian, sehingga dapat mengurangi tindakan curang yang dilakukan oleh murid. Oleh karena itu penulis mengangkat judul penelitian “Penerapan Algoritma Fisher-Yates Pada Computer Based Test Di Panda Mandarin Course”.

LANDASAN TEORI

Algoritma Fisher-Yates

Algoritma Fisher-Yates adalah sebuah algoritma yang menghasilkan permutasi acak dari satu set terbatas [2]. Hasil dari algoritma ini tidak akan berat sebelah jika diimplementasikan dengan benar, sehingga setiap permutasi memiliki kemungkinan yang sama. Metode Pengacakan Fisher-Yates ditemukan oleh Ronald Fisher dan Frank Yates. Metode ini pertama kali diperkenalkan pada tahun 1934 dan di revisi kembali pada tahun 1948 [3].

Gambar 1. Flowchart Metode Pengacakan Algoritma Fisher-Yates



Dari Gambar 1, diketahui bahwa algoritma Fisher-Yates diawali dengan

menentukan nilai n dalam mengacak urutan data. Kemudian tentukan angka acak x dimana x tidak boleh kurang dari 1 dan lebih dari n. Lalu tukar posisi x dengan angka terakhir pada range 1 sampai dengan n dan pindahkan angka x ke list array ditempat terpisah. Kemudian atur ulang nilai n dimana n = n-1, jika n bernilai 0 maka proses pengacakan dihentikan. Pada Gambar 1 terlihat bahwa pengacakan akan selesai jika seluruh array telah diacak [2].

Tabel 1. Analisis Pengacakan Urutan Soal

Range	Roll	Scratch	Result
		1,2,3,4,5	
1-5	4	1,2,3,5	4
1-4	2	1,3,5	2,4
1-3	3	1,3	5,2,4
1-2	2	1	3,5,2,4
			1,3,5,2,4

Tabel 1 menunjukkan penjelasan analisis pengacakan urutan soal dalam bentuk tabel. Terdapat 5 buah soal terurut yang akan diacak urutannya. Range adalah jumlah soal yang belum terpilih, Roll adalah sebuah soal acak yang terpilih dari jumlah soal yang ada, Scratch adalah daftar soal yang belum terpilih, dan Result adalah hasil permutasi dari seluruh soal yang didapatkan [3].

Computer Based Test

Computer Based Test adalah ujian yang telah terkomputerisasi sehingga dapat diatur dan diprogram sesuai kebutuhan. Pendidik dapat memonitor motivasi siswa, dan dapat mempersiapkan bahan tes dengan lebih berkualitas [4]. Computer Based Test merupakan ujian yang dilakukan menggunakan komputer dan dapat digunakan dalam lingkungan yang diawasi atau tidak diawasi [5].

Basis Data

Basis data adalah media penyimpanan data berupa tabel-tabel dan merupakan kumpulan dari field dan kolom [6]. Basis data merupakan sekumpulan data yang saling terkait yang diorganisasikan untuk memudahkan aktivitas memperoleh informasi [7].

METODE PENELITIAN

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Prototype*. Namun pada penelitian ini tidak terdapat tahapan evaluasi.

Komunikasi

Pada tahap ini penulis melakukan komunikasi dengan pihak Panda Mandarin *Course* mengenai kebutuhan perangkat lunak yang diinginkan. Peneliti juga melakukan pengumpulan data yang akan digunakan dalam pembuatan sistem. Melalui tahap ini didapat sebuah kebutuhan perangkat lunak, yaitu ujian berbasis komputer.

Perencanaan Cepat

Pada tahap ini penulis melakukan analisis dan perencanaan dari data-data dan informasi yang didapatkan pada tahap komunikasi dengan pihak Panda Mandarin *Course*.

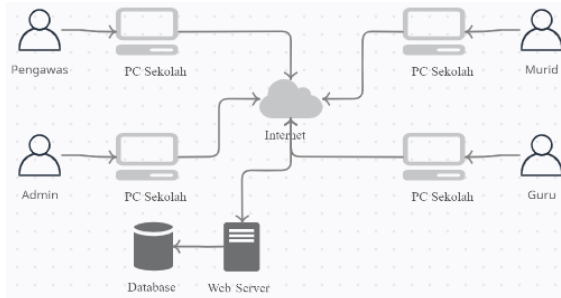
Pemodelan Perancangan Cepat

Pada tahap ini penulis melakukan perancangan desain program dari analisis sistem yang diajukan kepada pihak Panda Mandarin *Course*. Perancangan cepat yang dilakukan meliputi pembuatan arsitektur sistem, *use case diagram*, *activity diagram*, relasi antar tabel, dan penjelasan bagaimana sistem membatasi setiap kategori soal.

Arsitektur Sistem

Arksitektur sistem berikut menjelaskan tentang aktivitas yang perlu dilakukan oleh *user* didalam sistem untuk menjalankan sistem.

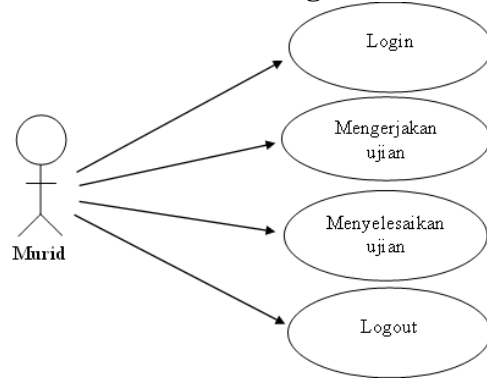
Gambar 2. Arsitektur Sistem



Use case Diagram

Use Case Diagram berikut menjelaskan tentang kebutuhan sistem dari sudut pandang murid.

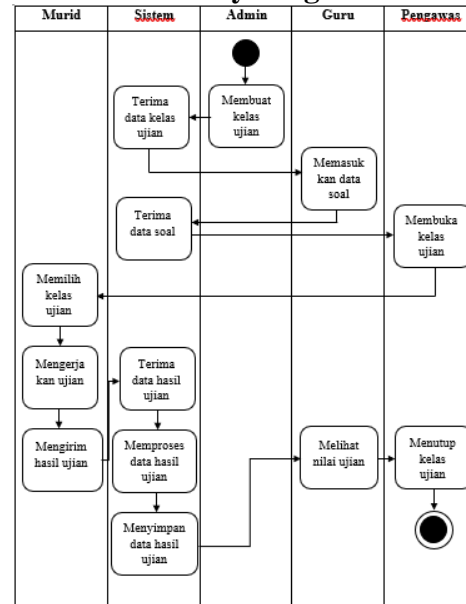
Gambar 3. Use case Diagram



Activity Diagram

Activity Diagram menjelaskan tentang aktivitas yang terjadi didalam sistem. Pertama admin akan membuat kelas ujian. Kemudian guru akan memasukkan data soal pada kelas ujian yang telah dibuat. Lalu pengawas akan membuka kelas ujian agar ujian dapat dikerjakan oleh murid. Kemudian murid akan memilih kelas ujian yang tersedia dan mengerjakan ujian. Saat sudah selesai mengerjakan, murid akan menyelesaikan dan mengirim data hasil ujian kepada sistem. Lalu sistem akan memproses dan menyimpan data hasil ujian. Kemudian guru dapat melihat nilai ujian. Kemudian pengawas dapat menutup kelas ujian.

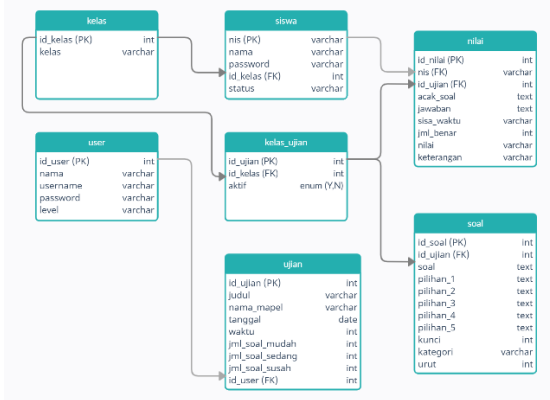
Tabel 2. Activity Diagram



Relasi Antar Tabel

Berikut ini adalah rancangan relasi antar tabel untuk sistem *computer based test* Panda Mandarin Course. Kemudian rancangan ini akan dibuat dalam bentuk *database*.

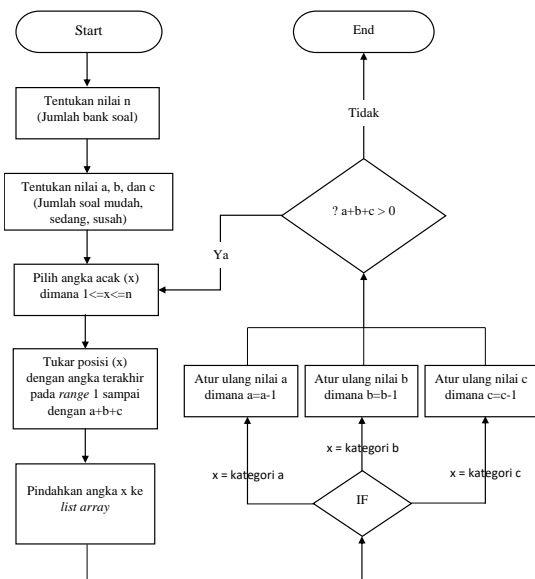
Gambar 4. Relasi Antar Tabel



Pembatasan Setiap Kategori Soal

Jumlah soal yang diberikan dapat dibatasi sesuai dengan keinginan guru. Pembatasan jumlah soal di dalam sistem dilakukan dengan cara membatasi setiap kategori soal pada saat admin membuat atau mengedit kelas ujian. Kemudian pembatasan kategori soal tersebut akan dijalankan ketika sistem akan mengambil soal secara acak dari *database*. *Flowchart* dari pembatasan setiap kategori soal dapat dilihat pada Gambar 5.

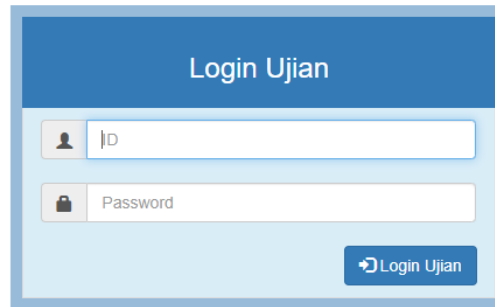
Gambar 5. Flowchart Sistem



HASIL DAN PEMBAHASAN Perancangan Halaman Login

Halaman *login* merupakan halaman awal yang terakses ketika pengguna masuk ke dalam *website*. Murid perlu memasukkan ID dan password untuk masuk ke halaman utama. Gambar 6 menunjukkan tampilan halaman *login computer based test* yang digunakan pada penelitian ini.

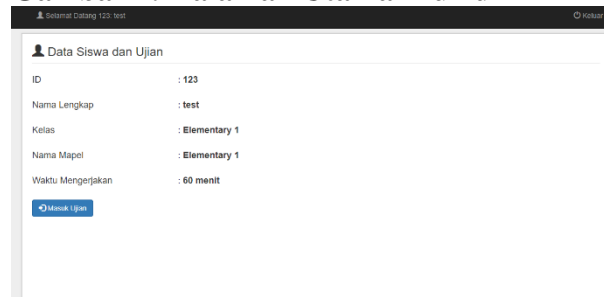
Gambar 6. Halaman Login Murid



Perancangan Halaman Murid

Setelah membuat halaman *login*, langkah selanjutnya adalah membuat halaman murid yang digunakan untuk mengerjakan ujian. Gambar 7 menunjukkan halaman utama yang terakses setelah murid melakukan *login*. Dari halaman ini murid dapat masuk dan mengerjakan ujian yang tersedia dengan cara menekan tombol masuk ujian.

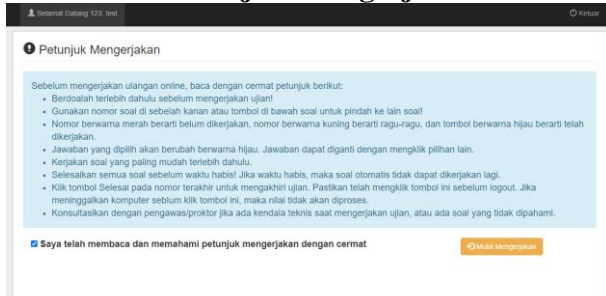
Gambar 7. Halaman Utama Murid



Gambar 8 menunjukkan menu yang ditampilkan oleh sistem jika murid menekan tombol masuk ujian. Sebelum mengerjakan ujian murid akan diberikan petunjuk untuk mengerjakan ujian agar murid dapat memahami sistem ujian dengan mudah. Kemudian murid perlu memberi ceklist pada kotak kecil di

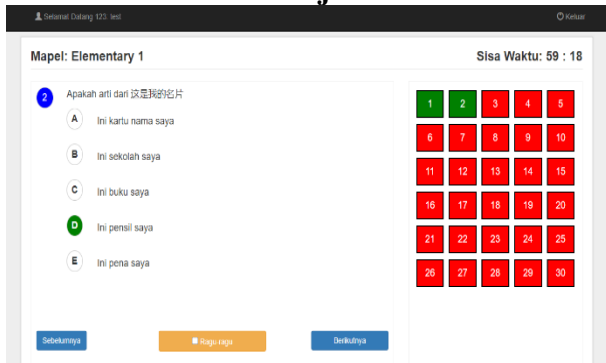
sebelah kiri kalimat “Saya telah membaca dan memahami petunjuk mengerjakan dengan cermat”. Lalu murid dapat menekan tombol mulai mengerjakan untuk masuk ke dalam ujian.

Gambar 8. Petunjuk Mengerjakan



Gambar 9 menunjukkan halaman ujian yang terakses setelah murid menekan tombol mulai mengerjakan. Murid akan diberikan soal beserta jawaban pada halaman ini. Terdapat beberapa hal yang dapat dilakukan oleh murid pada halaman ini, yaitu melihat sisa waktu ujian, menjawab pertanyaan, melihat soal sebelumnya dan berikutnya, menekan tombol ragu-ragu jika murid merasa jawaban yang dipilih belum benar agar murid dapat dengan mudah mencari soal tersebut setelah mengerjakan soal yang lain, dan memilih nomor soal untuk melihat soal pada nomor tersebut dengan cara menekan tombol nomor soal pada bagian kanan halaman.

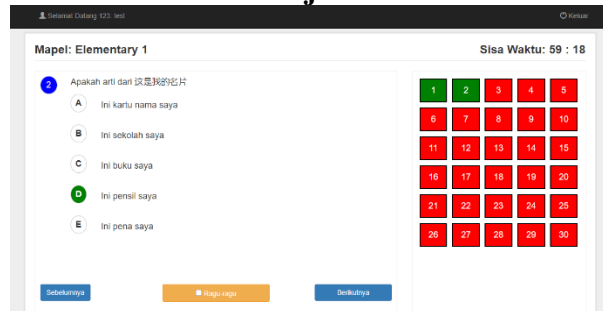
Gambar 9. Halaman Ujian



Gambar 10 menunjukkan menu yang ditampilkan oleh sistem jika murid telah menjawab semua soal dan menekan tombol selesai pada bagian bawah halaman. Murid akan diberikan notifikasi oleh sistem sebelum menyelesaikan ujian untuk memastikan semua

soal telah dikerjakan. Jika murid sudah yakin telah mengerjakan semua soal maka murid dapat menekan tombol selesai.

Gambar 10. Selesai Ujian



Implementasi Algoritma Fisher-Yates

Setelah semua halaman pada *Computer Based Test* telah dibuat maka langkah selanjutnya adalah mengimplementasikan algoritma *Fisher-Yates* ke dalam *Computer Based Test* agar soal dan jawaban dapat teracak. Tahap implementasi dibagi menjadi dua langkah, yaitu:

1. Implementasi Algoritma Fisher-Yates Pada Soal

Langkah pertama ialah memasukkan pengkodean algoritma *Fisher-Yates* untuk mengacak soal pada sistem. Pengkodean tersebut dapat dilihat pada Gambar 11.

Gambar 11. Implementasi Algoritma Fisher-Yates Pada Soal

```

3 <?php
4 // fisher-yates shuffle soal
5 function MyShuffle2(&$arr_soal) {
6
7     for($i = 0; $i < sizeof($arr_soal); ++$i) {
8         $r = rand(0, $i);
9         $tmp = $arr_soal[$i];
10        $arr_soal[$i] = $arr_soal[$r];
11        $arr_soal[$r] = $tmp;
12    }
13 };
    
```

Kemudian hasil pengkodean tersebut dijalankan saat soal ujian akan diambil oleh sistem, seperti pada Gambar 12.

Gambar 12. Menjalankan Algoritma Fisher-Yates Pada Soal

```

72 $arr_soal = array_merge($arr_soal1,$arr_soal2,$arr_soal3);
73 $arr_jawaban = array_merge($arr_jawaban1,$arr_jawaban2,$arr_jawaban3);
74
75 MyShuffle2($arr_soal);
76 $acak_soal = implode(" ", $arr_soal);
77 $jawaban = implode(" ", $arr_jawaban);
    
```

2. Implementasi Algoritma Fisher-Yates Pada Jawaban

Langkah kedua ialah memasukkan pengkodean algoritma Fisher-Yates untuk mengacak jawaban pada sistem. Pengkodean tersebut dapat dilihat pada Gambar 13.

Gambar 13. Implementasi Algoritma Fisher-Yates Pada Jawaban

```
14 >
15 <?php |
16 // fisher-yates shuffle jawaban
17 function MyShuffle(&$arr_pilihan) {
18
19     for($i = 0; $i < sizeof($arr_pilihan); ++$i) {
20         $r = rand(0, $i);
21         $tmp = $arr_pilihan[$i];
22         $arr_pilihan[$i] = $arr_pilihan[$r];
23         $arr_pilihan[$r] = $tmp;
24     }
25 };
26 >
27 <?php
```

Kemudian hasil pengkodean tersebut dijalankan saat jawaban akan ditampilkan oleh sistem, seperti pada Gambar 14.

Gambar 14. Menjalankan Algoritma Fisher-Yates Pada Jawaban

```
117 //6 Membuat array pilihan dan mengacak pilihan
118 $arr_pilihan = array();
119 $arr_pilihan[] = array("no" => 1, "pilihan" => $rsoal['pilihan_1']);
120 $arr_pilihan[] = array("no" => 2, "pilihan" => $rsoal['pilihan_2']);
121 $arr_pilihan[] = array("no" => 3, "pilihan" => $rsoal['pilihan_3']);
122 $arr_pilihan[] = array("no" => 4, "pilihan" => $rsoal['pilihan_4']);
123 $arr_pilihan[] = array("no" => 5, "pilihan" => $rsoal['pilihan_5']);
124 MyShuffle($arr_pilihan);
125
```

Hasil Implementasi Algoritma Fisher-Yates

Setelah selesai mengimplementasikan algoritma Fisher-Yates pada soal dan jawaban kedalam computer based test, maka langkah selanjutnya dilakukan pengecekan langsung kedalam program untuk memastikan apakah soal dan jawaban yang diterima setiap murid dapat teracak. Pada Tabel 3 dan Tabel 4 dapat terlihat hasil pengacakan soal yang diterima oleh setiap murid yang telah mengerjakan ujian. Setiap murid akan mendapatkan 30 soal secara acak dari 90 soal yang ada pada bank soal. 30 soal tersebut terdiri dari 10 soal mudah, 10 soal sedang, dan 10 soal susah. Kategori soal mudah adalah urutan soal dari nomor 1 sampai 30 dan tanda berwarna merah adalah soal mudah yang terpilih pada ujian setiap siswa. Kategori soal sedang adalah urutan soal nomor 31 sampai 60

dan tanda berwarna kuning adalah soal sedang yang terpilih pada ujian setiap siswa. Kategori soal susah adalah urutan soal dari nomor 61 sampai 90 dan tanda berwarna hijau adalah soal susah yang terpilih pada ujian setiap siswa. Pada kasus ini terdapat 30 murid yang mengerjakan ujian Elementary 1 dengan 90 bank soal. Setiap murid akan mendapatkan 30 soal dengan waktu mengerjakan selama 60 menit.

Tabel 3. Hasil Pengacakan Soal Siswa 1 Sampai 15

Table with 30 columns (Claryn, Ellen, Federica, Sharon, Jovanna, Eunice, AnFaith, Alvin Wid, Gabby, Kenzo, Michelle, Love Divi, Villa Aurd, Gerald, Jesslyn, Se) and 90 rows of data. Each cell contains a number from 1 to 90, color-coded to represent difficulty levels: red for easy (1-30), yellow for medium (31-60), and green for difficult (61-90).

Tabel 4. Hasil Pengacakan Soal Siswa 16 Sampai 30

Table with 16 columns (Soal No. 1-16) and 16 rows (Siswa 1-16). Each cell contains a number representing the question number selected by the student. The table shows a pattern where certain question numbers are selected more frequently by certain students.

Dari tabel tersebut dapat terlihat bahwa murid bernama Claryn mendapat soal nomor 2, 11, 14, 17, 18, 22, 25, 27, 28, 30, 34, 38, 39, 40, 41, 47, 51, 52, 53, 54, 62, 64, 71, 73, 77, 82, 84, 85, 88, 89 sedangkan murid bernama Eillen mendapat soal nomor 5, 8, 9, 11, 12, 16, 21, 22, 26, 27, 35, 38, 42, 45, 46, 47, 52, 53, 59, 60, 62, 64, 67, 69, 70, 73, 77, 78, 82, 88. Dari kedua murid tersebut dapat dilihat bahwa mereka mendapat 13 soal yang sama, dan dari 13 soal yang sama tersebut terdapat 4 soal yang nomor urut nya juga sama. Hal ini berarti terjadi kemunculan soal yang sama sebanyak 44% atau kurang dari 44% dan terjadi kemunculan soal yang sama pada nomor urut yang sama sebanyak 14% atau kurang dari 14% yang mengindikasikan kemungkinan terjadinya kemunculan soal yang sama di waktu yang

bersamaan pada 2 orang murid sangatlah kecil. Dari Tabel 4 juga dapat dilihat persentase kemunculan setiap soal dan dapat diketahui rata-rata kemunculan soal adalah sebesar 3,33%.

PENUTUP Kesimpulan

Berdasarkan hasil dan pembahasan pada penelitian ini, maka didapat beberapa kesimpulan sebagai berikut :

- 1. Algoritma Fisher-Yates berhasil diimplementasikan pada Computer Based Test di Panda Mandarin Course.
2. Algoritma Fisher-Yates dapat melakukan acak soal dan jawaban berdasarkan tingkat kesulitan.
3. Perangkat lunak ujian Computer Based Test dapat mengurangi tindakan kecurangan yang dilakukan oleh murid.

Saran

Penelitian ini tentunya masih memiliki kekurangan serta keterbatasan dan jauh dari kata sempurna. Berikut beberapa saran yang bisa dilakukan untuk menyempurnakan penelitian ini :

- 1. Penelitian ini hanya menggunakan algoritma Fisher-Yates untuk mengacak soal dan jawaban, perlu adanya penelitian lain yang menggunakan algoritma pengacakan yang berbeda.
2. Bahasa pemrograman yang digunakan pada penelitian ini adalah PHP dan HTML diharapkan ada penelitian lain yang menggunakan bahasa pemrograman yang berbeda.
3. Penelitian ini tidak dapat melihat hasil rekaman soal dan jawaban yang dikerjakan oleh murid, diharapkan ada penilitan berikutnya yang dapat merekam soal dan jawaban yang dikerjakan oleh murid.
4. Pada penelitian ini tidak terdapat menu latihan yang dapat dikerjakan berulang-ulang, diharapkan ada penelitian berikutnya yang dapat membuat menu latihan untuk dikerjakan berulang-ulang.

DAFTAR PUSTAKA

- [1] S. Susilowati and T. Hidayat, “Rancang Bangun Sistem Informasi Ujian Online (Studi Kasus Pada SMAN 58 Jakarta),” *J. Tek. Komput.*, vol. 4, no. 1, pp. 30–36, 2018, [Online]. Available: <http://bsnp-indonesia.org>.
- [2] Ekojono, D. A. Irawati, L. Affandi, and A. N. Rahmanto, “Penerapan Algoritma Fisher-Yates Pada Pengacakan Soal Game Aritmatika,” *Pros. SENTIA 2017 – Politek. Negeri Malang*, vol. 9, no. ISSN: 2085-2347, pp. 101–106, 2017, [Online]. Available: <http://sentia.polinema.ac.id/index.php/SENTIA2017/article/viewFile/237/225>.
- [3] V. Asih, A. Saputra, and R. T. Subagio, “Penerapan Algoritma Fisher Yates Shuffle Untuk Aplikasi Ujian Berbasis Android,” *J. Digit*, vol. 10, no. 1, p. 59, 2020, doi: 10.51920/jd.v10i1.156.
- [4] P. D. A. Pamungkas, “Computer Based Test (CBT) Pada Sekolah Tinggi Tarakanita Jakarta Menggunakan Metode Computerized Fixed-Form Test (CFT),” *J. Ilm. Teknol. Inf. Terap.*, vol. IV, no. 1, pp. 54–61, 2017.
- [5] B. Tjahjo, “Computer Based Test Fakultas Ilmu Komputer Pendahuluan Tujuan & Manfaat Teori Dasar Metode Penelitian,” vol. V, no. 2, pp. 25–34, 2018.
- [6] A. Kristian, “Sistem Informasi Rencana Pembelajaran Semester Berbasis Website pada IIB Darmajaya,” 2020.
- [7] A. R. Adfa, “Perancangan sistem informasi museum lampung berbasis website,” 2019.