

Analisis Hasil Uji Algoritma Fisher-Yates pada Aplikasi Simulasi CAT CPNS

Danny Kriestanto¹, Azhari Hilmi²

Jurusan Teknik Informatika STMIK AKAKOM

Jl Raya Janti no 143 Karang Jambe, Bantul, Yogyakarta, 55198

danny@akakom.ac.id, cacakmati21@gmail.com

INTISARI

Calon pegawai negeri sipil yang akan mengikuti ujian masuk calon pegawai negeri sipil akan dihadapkan dengan banyak pertanyaan ujian. Diperlukan suatu aplikasi yang dapat membantu para calon pegawai negeri sipil dalam mengasah kemampuannya diperlukan sebuah simulasi ujian masuk. Simulasi tersebut haruslah dapat menampilkan pertanyaan-pertanyaan yang dimunculkan secara acak.

Dalam proses pengacakan bilangan dengan menggunakan random, kemungkinan munculnya bilangan yang sama dapat terjadi. Salah satu algoritma yang dapat digunakan untuk menampilkan bilangan acak yang tidak berulang adalah algoritma Fisher-Yates.

Algoritma Fisher-Yates tidak memungkinkan munculnya nomor indeks soal yang sama pada peserta simulasi yang sama, namun diharapkan dapat memberikan nomor indeks soal yang berbeda pada peserta yang berbeda pula.

Adapun hasil penelitian ini membuktikan bahwa kemungkinan munculnya nomor indeks yang sama pada nomor soal yang sama pula bergantung dari banyaknya indeks soal dan banyaknya soal yang dipanggil dari bank soal tersebut. Semakin banyak indeks bank soal dan semakin sedikit jumlah soal yang diambil, semakin kecil pula kemungkinan soal yang berulang.

Kata kunci— CAT PNS, Bilangan Acak, Pengacakan Fisher-Yates, PHP, Aplikasi Web

ABSTRACT

In order to pass one of the criteria to become a civil servant, the candidates must face a lot of exam questions. An application to help these candidates is highly required. This simulation must be able to show the questions randomly.

In randomize numbers, the probability of a number to be shown again is quite high. One of the algorithms to handle this problem is Fisher-Yates.

In Fisher-Yates Algorithm, there is not such thing as a duplicate number for the same exam candidate, but this research focus is to find a correlation between index number and the data in database.

The result showed that the reoccurrence of the same index number on the same question number depends on the amount of index number and the number of data queried. The amount the data in the database must be big and the index number must be small in order to reduce the probability of reoccurrence of same index number among the examinees.

Keywords— CAT PNS, Random Number, Fisher Yates Shuffle, PHP, Web Application

I. PENDAHULUAN

A. Latar Belakang

Tak bisa dipungkiri bahwa setiap ada lowongan CPNS banyak yang siap bersaing untuk bisa lulus sebagai seorang PNS. Untuk menjadi seorang PNS salah satu tantangan adalah pada tes CAT CPNS. Pada sebuah tes *Computer Assisted Tests* (CAT) diperlukan adanya pengacakan soal antara satu peserta dengan peserta yang lain. Selain itu, diperlukan juga pengacakan soal agar soal yang sama tidak

muncul lebih dari satu kali. Algoritma Fisher-Yates dapat memastikan bahwa bilangan yang diacak muncul hanya sekali.

Fisher-Yates merupakan sebuah algoritma untuk menghasilkan permutasi acak dari suatu himpunan berhingga [1]. Jika diimplementasikan dengan benar maka hasil dari algoritma ini tidak akan berat sebelah sehingga setiap permutasi memiliki kemungkinan yang sama.

Algoritma Fisher-Yates terbagi atas dua macam metode, yakni orisinal dan modern, namun dalam penelitian ini akan menggunakan metode modern yang digunakan oleh Richard Durstenfeld. Metode ini lebih optimal dibandingkan metode orisinal karena angka terakhir yang ditukar dengan posisi angka yang nilainya ditarik keluar. Dengan demikian, ruang dan waktu kompleksitas algoritmanya optimal [2].

Adapun batasan yang digunakan dalam penelitian ini adalah:

1. Pada level aplikasi akan digunakan 2 jenis tes pada bank soal, yakni: Tes Inteligensi Umum (TIU) dan Tes Wawasan Kebangsaan (TWK)
2. Bentuk soal adalah pilihan tanpa ada esai.
3. Simulasi dilakukan berbasis web.
4. Pengguna umum tidak perlu melakukan login dan tidak ada penyimpanan riwayat hasil simulasi.

B. Tinjauan Pustaka

Referensi [3] menerapkan algoritma Fisher-Yates pada game untuk mengenali serangga. Penelitian tersebut melihat persentase kepuasan dan kemudahan aplikasinya.

Referensi [4] menggunakan algoritma Fisher-Yates untuk permainan aritmetika sebagai pengacak soal.

Referensi [5] menggunakan algoritma Fisher-Yates yang mirip dengan [4] namun dikhususkan untuk anak-anak. Pengujian aplikasi tidak dilakukan dengan menguji hasil implementasi algoritma.

Referensi [6] menggunakan Fisher-Yates sebagai pengacak pertanyaan permainan labirin. Penelitian tersebut membandingkan.

Referensi [7] menggunakan Fisher-Yates untuk mengacak indeks dengan menampilkan 10 soal. Namun penelitian ini tidak melakukan perbandingan untuk beberapa pengguna yang berbeda sekaligus.

Referensi [8] menerapkan Fisher-Yates untuk pembelajaran tenses pada bahasa Inggris. Pengacakan dilakukan pada soal-soal latihan.

Referensi [9] membuat aplikasi tes IQ dengan menerapkan algoritma Fisher-Yates. Hasil aplikasi hanya diuji tiga kali menggunakan dua komputer secara bersamaan.

C. Tujuan Penelitian

Adapun penelitian dalam naskah ini berfokus untuk melihat hasil pengacakan bilangan dengan Fisher-Yates dimana angka indeks soal yang dipanggil tidak muncul di nomor urut soal yang sama pada peserta lain.

Penelitian ini juga akan menyelidiki apakah ada pengaruh antara jumlah indeks dengan

banyaknya soal yang ditampilkan kepada pengguna.

Selain itu, seberapa banyak jumlah bank soal yang kira-kira dibutuhkan agar dapat mengurangi terjadinya pengulangan indeks.

D. Algoritma Fisher-Yates

Metode dasar yang digunakan untuk menghasilkan suatu permutasi acak untuk angka 1 sampai N adalah sebagai berikut:

1. Tuliskan angka dari 1 sampai n (range).
2. Pilih sebuah angka acak x diantara 1 sampai dengan jumlah n nya.
3. Pindahkan angka acak x ke hasil
4. Letakan angka terakhir pada posisi angka acak x sebelumnya.
5. Ulangi langkah 2, 3 dan 4 sampai semua range terpenuhi.
6. Urutan angka yang dituliskan pada langkah 3 adalah hasil dari permutasi acak.

Tabel I menunjukkan cara pengerjaan algoritma Fisher-Yates, dimana *Range* adalah panjang bilangan yang belum terpilih, *Random* merupakan angka acak yang dibangkitkan hanya di dalam *Range*, *Scratch* merupakan urutan angka yang belum dipilih, dan *Result* merupakan urutan hasil permutasi yang diperoleh.

Tabel I.
Contoh Pengerjaan Algoritma Fisher-Yates

Range	Random	Scratch	Result
1-8		1 2 3 4 5 6 7 8	
1-7	6	1 2 3 4 5 8 7	6
1-6	2	1 7 3 4 5 8	2 6
1-5	6	1 7 3 4 5	8 2 6
1-4	1	5 7 3 4	1 8 2 6
1-3	3	5 7 4	3 1 8 2 6
1-2	3	5 7	4 3 1 8 2 6
1	1	7	5 4 3 1 8 2 6
	1		7 5 4 3 1 8 2 6
	Hasil Pengacakan		7 5 4 3 1 8 2 6

Aplikasi yang dibuat hanya berupa latihan bagi calon peserta CAT CPNS, oleh karena itu bentuk soal yang disajikan hanyalah berupa pilihan ganda.

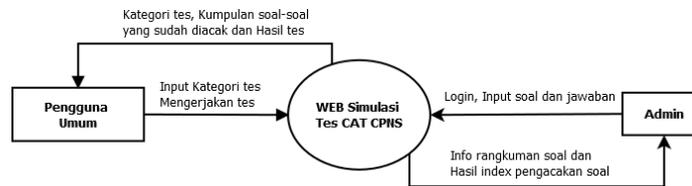
II. METODOLOGI PENELITIAN

Penelitian dilakukan dengan mengimplementasikan algoritma pengacakan Fisher-Yates dalam bentuk skrip PHP. Hasilnya kemudian diterapkan dalam aplikasi sederhana latihan CAT untuk CPNS. Hasil uji coba kemudian ditampilkan ke dalam bentuk tabel.

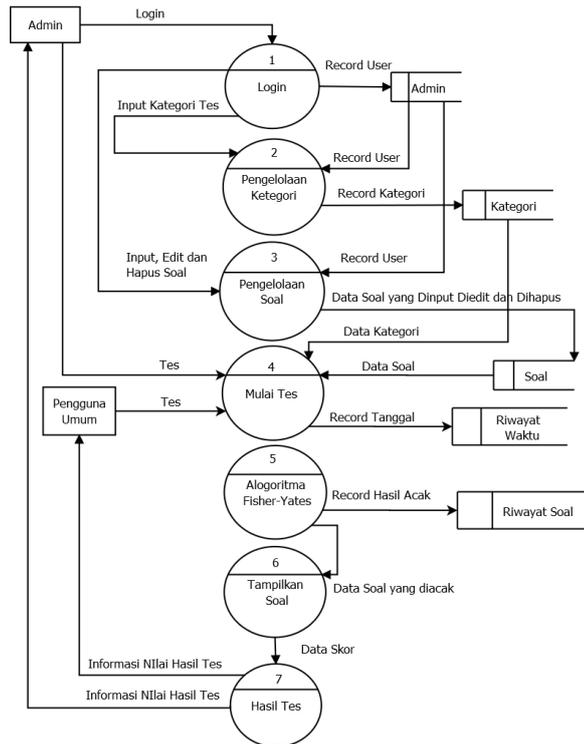
User yang dapat menggunakan aplikasi ini terdiri atas 2 macam, yakni: Admin dan Pengguna Umum. Admin bertugas untuk mengelola data pada website, termasuk di dalamnya memasukkan, mengedit, dan menghapus data-data. Sedangkan Pengguna

Umum adalah pemakai aplikasi ini. Pengguna Umum dapat bebas menggunakan aplikasi ini tanpa perlu melakukan login.

Adapun diagram alir data untuk penelitian ini adalah seperti yang tampil pada Diagram Konteks pada Gambar 1.



Gambar 1. Diagram Konteks Simulasi CAT CPNS

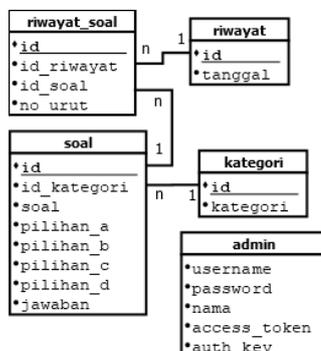


Gambar 2. Diagram Alir Data Simulasi CAT CPNS

Diagram konteks pada Gambar 1 terlihat bahwa terdapat 2 entitas luar yakni pengguna umum dan Admin sebagai pengelola aplikasi. Diagram ini kemudian dijabarkan dalam bentuk Diagram Alir Data level 1 seperti yang tampak pada Gambar 2.

Inti diagram pada Gambar 2 terletak pada proses 5 dimana algoritma Fisher-Yates diterapkan sebagai pengacak. Dan perancangan basis data untuk aplikasi ini dengan mengacu pada konsep yang ditulis pada referensi [10], dapat dilihat pada Gambar 3.

Adapun detail dari skema tersebut adalah seperti yang terlihat pada Tabel II hingga Tabel VII berikut.



Gambar 3. Desain Basis Data Simulasi CAT CPNS

Tabel II.

Struktur Tabel Admin

No	Nama Field	Tipe	Panjang
1	Id_admin	Int	6
2	Username	Varchar	15
3	Password	Varchar	25
4	aces_token	Varchar	128
5	auth_key	Varchar	128

Tabel Admin yang ditunjukkan oleh Tabel II berfungsi untuk menyimpan data-data admin.

Aplikasi ini memungkinkan admin yang lebih dari satu orang. Setiap admin diberi token dan kunci tersendiri.

Tabel III.
Struktur Tabel Kategori

No	Nama Field	Tipe	Panjang
1	Id	Int	11
2	kategori	Varchar	30

Tabel III menunjukkan struktur tabel Kategori soal. Desain ini memungkinkan adanya penambahan kategori baru apabila diperlukan.

Tabel IV.
Struktur Tabel Soal

No	Nama Field	Tipe	Panjang
1	Id	Int	11
2	Id_kategori	Int	11
3	soal	Text	
4	pilihan_a	Text	
5	pilihan_b	Text	
6	pilihan_c	Text	
7	pilihan_d	Text	
8	jawaban	Varchar	1

Struktur tabel Soal yang ditunjukkan pada Tabel IV memungkinkan pengembangan aplikasi untuk pengacakan urutan jawaban. Dalam penelitian ini pengacakan urutan jawaban tidak diimplementasikan.

Tabel V.
Struktur Tabel Riwayat

No	Nama Field	Tipe	Panjang
1	id	Int	11
2	tanggal	Date	

Fungsi dari tabel Riwayat pada Tabel V adalah untuk menyimpan riwayat waktu tes. Tabel ini dapat dimodifikasi pada pengembangan selanjutnya untuk menyimpan skor latihan, apabila pengguna umum dimungkinkan untuk melakukan login terlebih dahulu.

Tabel VI.
Struktur Tabel Riwayat Soal

No	Nama Field	Tipe	Panjang
1	Id	Int	11
2	id_riwayat	Int	11
3	Id_soal	Int	11
4	no_urut	Int	11

Fungsi utama dari tabel Riwayat Soal yang strukturnya terlihat pada Tabel VII adalah untuk menyimpan nilai indeks soal. Sifat dari tabel ini hanyalah sementara untuk setiap pengguna dan dipergunakan untuk penampung sementara soal-soal yang akan ditampilkan kepada pengguna.

III. HASIL DAN PEMBAHASAN

A. Hasil

Dari sisi pengguna umum, tampilan yang muncul pada ketika melakukan simulasi menunjukkan beberapa informasi, seperti batas waktu menjawab pertanyaan, jumlah soal yang diujikan, banyak soal yang telah dijawab, banyaknya soal yang belum dijawab, dan tombol untuk selesai ujian. Tampilannya dapat dilihat seperti pada Gambar 4.



Gambar 4. Tampilan Menu Simulasi Ujian

Pengguna umum dapat melihat hasil ujian ketika simulasi ujian selesai. Adapun tampilannya seperti yang tampak pada Gambar 5.

Tugas utama Admin adalah menambahkan bank soal. Bentuk soal yang dapat ditambahkan hanyalah Tes Intelegensi Umum (TIU) dan Tes Wawasan Kebangsaan (TWK). Tampilannya terlihat seperti yang ditunjukkan pada Gambar 6.

Informasi Simulasi	
Jenis Ujian	: Intelegensi Umum
Waktu Ujian	: 10 menit
Waktu Mulai	: 11:42 27-08-2018
Waktu Selesai	: 11:46 27-08-2018
Informasi Hasil Ujian	
Nilai Intelegensi Umum	: 90

Gambar 5. Tampilan Hasil Tes



Gambar 6. Halaman Master Soal

Terdapat fitur tambahan untuk mencatat riwayat pengacakan angka, yang dapat dilihat pada Gambar 7 berikut ini.

Showing 1-10 of 10 items.

#	Id Soal	No Urut
1	18	1
2	1	2
3	7	3
4	12	4
5	41	5
6	19	6
7	6	7
8	4	8
9	11	9
10	17	10

Gambar 7. Halaman Riwayat Pengacakan

Hasil pengacakan tersebut ditampilkan hasil pengacakan simulasi tes yang sebelumnya telah ditampilkan.

Dari Gambar 7 terlihat bahwa hasil pengacakan dengan menggunakan algoritma Fisher-Yates tidak terdapat angka indeks yang berulang (kolom Id Soal).

B. Pembahasan

Proses pengujian algoritma Fisher-Yates dilakukan sebanyak empat kali pada 10 peserta yang mengakses aplikasi ini dalam waktu yang hampir bersamaan.

Proses pengujiannya adalah sebagai berikut:

- Pengujian pertama: menampilkan 10 soal dari 40 soal yang terdapat di dalam bank soal
- Pengujian kedua: menampilkan 20 soal dari 40 soal yang terdapat di dalam bank soal
- Pengujian ketiga: menampilkan 10 soal dari 100 soal yang terdapat di dalam bank soal
- Pengujian keempat: menampilkan 20 soal dari 100 soal yang terdapat di dalam bank soal

Adapun hasil dari Pengujian dapat dilihat pada Tabel VII hingga Tabel X. Adapun yang diberi warna adalah angka acak yang sama pada nomor soal yang sama pula.

Tabel VII.
Hasil Pengujian Fisher-Yates Pertama – Uji Kemunculan Indeks yang Sama pada No Soal yang Sama

Peserta	Soal No-									
	1	2	3	4	5	6	7	8	9	10
P1	37	11	20	22	40	29	30	3	36	33
P2	19	27	6	13	4	2	34	40	11	17
P3	30	28	16	35	23	4	7	24	10	5
P4	28	34	4	27	37	1	15	10	7	36
P5	25	13	38	18	31	15	12	23	21	10
P6	38	14	11	24	12	37	32	1	19	23
P7	29	7	18	16	10	23	20	4	39	32
P8	40	38	23	2	18	24	10	22	13	26
P9	38	36	12	31	9	7	20	39	14	34
P10	16	5	40	34	10	8	15	23	26	37

Pada pengujian pertama (Tabel VII), setelah melakukan pengacakan sebanyak 10 kali pada

10 peserta yang berbeda, tampak pada tabel bahwa hanya terdapat satu nomor indeks yang

berulang pada nomor soal yang sama (nomor 10 pada tabel).

Berdasarkan hasil uji pada Tabel VII, persentase banyaknya indeks yang muncul

berulang pada nomor urut soal yang sama adalah sebesar 20%.

Tabel VIII.
Hasil Pengujian Fisher-Yates Kedua – Uji Kemunculan Indeks yang Sama pada No Soal yang Sama

Peserta	Soal No-																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	33	40	36	25	20	2	27	17	4	15	23	39	30	3	38	16	28	26	8	1
P2	19	35	10	6	40	5	3	23	21	13	31	17	26	8	20	34	11	25	38	39
P3	21	25	39	5	19	28	9	4	10	13	6	33	22	32	24	34	26	37	17	36
P4	16	25	2	20	41	22	1	10	34	24	38	26	12	33	3	28	31	9	11	15
P5	11	4	33	32	29	14	25	39	41	12	2	1	5	36	30	10	9	26	16	8
P6	18	1	24	10	19	35	4	23	41	16	11	31	38	33	28	13	40	27	7	30
P7	3	23	19	37	1	36	35	21	38	39	20	22	9	24	5	31	40	32	12	2
P8	35	16	1	4	13	14	9	40	28	22	32	26	33	17	2	34	7	20	37	25
P9	34	31	23	18	19	12	3	5	26	11	7	10	16	2	21	22	32	8	25	27
P10	5	19	31	28	11	29	25	17	24	36	23	27	21	18	1	26	30	32	2	10

Pada pengujian kedua (Tabel VIII), setelah melakukan pengacakan sebanyak 20 kali dengan jumlah indeks yang sama, dengan jumlah soal sebanyak 20, dapat dilihat bahwa pada nomor soal yang sama terdapat 9 soal yang berulang antara peserta satu dengan yang lainnya. Sehingga, dalam hasil uji ini tingkat kemunculan bilangan yang sama pada nomor yang sama adalah sebesar 45%.

Pengujian ketiga (Tabel IX), setelah dilakukan pengacakan sebanyak 10 kali terhadap 100 index soal dan mengambil 10 nomor soal, hasil menunjukkan bahwa ada indeks soal yang berulang pada peserta 3 dan peserta 4, serta peserta 2 dan peserta 5. Dengan demikian, dari hasil pengacakan 100 indeks terdapat 20% kemungkinan keluarnya indeks soal yang sama.

Tabel IX.
Hasil Pengujian Fisher-Yates Ketiga – Uji Kemunculan Indeks yang Sama pada No Soal yang Sama

Peserta	Soal No-									
	1	2	3	4	5	6	7	8	9	10
P1	15	59	62	37	84	61	87	41	39	16
P2	41	53	56	96	60	26	70	65	68	86
P3	17	76	83	4	39	35	58	55	77	21
P4	54	14	1	4	48	50	32	16	91	7
P5	45	31	66	67	9	75	5	46	68	10
P6	69	77	44	11	21	37	98	40	85	97
P7	47	51	49	60	62	94	23	56	78	99
P8	25	55	33	23	75	12	86	26	59	89
P9	4	16	8	21	87	91	33	70	73	64
P10	62	67	55	44	53	77	85	71	41	2

Pengujian keempat (Tabel X), setelah dilakukan pengacakan 10 kali terhadap 100 index soal dengan mengambil 20 soal, tabel tersebut menunjukkan adanya 3 kali munculnya indeks soal yang sama.

Kemunculan nilai indeks pada hasil uji ini tidak signifikan jika dibandingkan hasil uji yang terlihat pada Tabel VIII dengan menggunakan jumlah soal yang sama.

Persentase kemunculan pada kasus ini adalah sebesar 15%.

Jika diperhatikan dengan saksama, kemunculan bilangan berulang pada posisi yang sama sebagian besar terletak bersebelahan. Dalam arti, soal yang sama di urutan yang sama terdapat pula di peserta lain yang bersebelahan.

Tabel X.

Hasil Pengujian Fisher-Yates Keempat – Uji Kemunculan Indeks yang Sama pada No Soal yang Sama

Peserta	Soal No-																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	47	93	92	43	20	16	69	25	80	28	42	79	45	55	82	52	44	19	32	76
P2	2	74	37	86	64	14	43	53	11	46	1	82	76	31	22	57	7	56	62	81
P3	31	45	70	29	64	51	46	98	27	93	68	77	17	13	20	21	4	74	75	32
P4	74	49	68	34	13	63	97	22	95	30	94	86	64	24	84	66	26	16	3	78
P5	43	6	69	37	22	99	83	39	58	57	21	65	94	60	59	18	23	75	53	29
P6	90	10	16	56	25	44	57	13	78	22	26	20	68	98	7	62	5	75	59	1
P7	72	70	6	51	78	18	31	36	21	71	15	81	84	96	76	67	83	12	5	89
P8	11	46	91	92	4	77	16	25	22	38	83	43	37	7	62	53	52	80	87	42
P9	39	35	87	59	50	6	58	24	51	52	61	67	28	34	44	22	62	64	86	68
P10	52	83	48	82	8	61	100	7	85	19	14	42	13	20	44	95	25	37	65	96

Hasil uji coba yang ditunjukkan oleh Tabel VII hingga Tabel X menunjukkan bahwa untuk jumlah 10 soal, baik dengan banyak bank soal sejumlah 40 soal maupun dengan bank soal sebanyak 100 soal, tidak ada perbedaan. Namun, perbedaan signifikan terlihat ketika jumlah soal yang diambil sebanyak 20 buah soal; dimana persentase kemunculan kembali sebesar 45% untuk 10 soal, berkurang hingga 15% untuk 20 soal. Dengan demikian, dapat disimpulkan bahwa jumlah soal di dalam bank soal juga dapat menentukan besarnya persentase muncul kembalinya indeks soal yang sama, walaupun jumlah bank soal tersebut hanya merupakan salah satu kemungkinan yang ada. Faktor pembangkit bilangan acak dapat pula menjadi salah satu penyebabnya.

Jika dilihat dengan lebih saksama, kemunculan indeks yang sama, baik pada hasil

uji dengan 10 soal maupun 20 soal dengan letak nomor soal yang berbeda, banyak terjadi. Jika Tabel VII dan Tabel IX dibandingkan, akan tampak seperti yang terlihat pada Tabel XI dan Tabel XII. Penataan tabel ini dimaksudkan agar supaya isi dari kedua tabel lebih mudah terlihat perbedaannya.

Tabel XI menunjukkan frekuensi kemunculan bilangan hasil acak. Dari sampel indeks yang ditandai, terlihat dengan jelas bahwa frekuensi kemunculan bilangan yang sama cukup banyak, seperti angka 10 muncul hingga 6 kali, angka 4 yang muncul hingga 4 kali, angka 23 yang muncul hingga lima kali, angka 18 yang muncul tiga kali, dan seterusnya. Hal ini disebabkan Fisher-Yates hanya dapat mengacak hingga 40 bilangan.

Tabel XI.

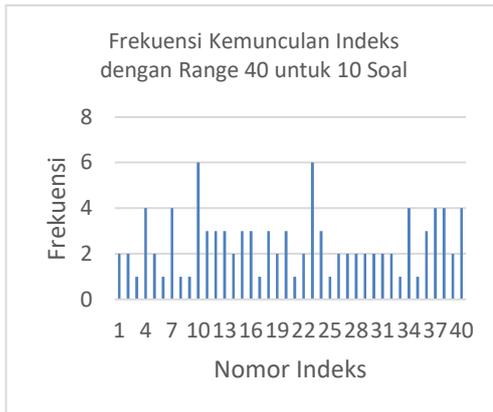
Hasil Pengujian Fisher-Yates Pertama – Uji Kemunculan Indeks yang Sama pada No Soal Berbeda

Peserta	Soal No-									
	1	2	3	4	5	6	7	8	9	10
P1	37	11	20	22	40	29	30	3	36	33
P2	19	27	6	13	4	2	34	40	11	17
P3	30	28	16	35	23	4	7	24	10	5
P4	28	34	4	27	37	1	15	10	7	36
P5	25	13	38	18	31	15	12	23	21	10
P6	38	14	11	24	12	37	32	1	19	23
P7	29	7	18	16	10	23	20	4	39	32
P8	40	38	23	2	18	24	10	22	13	26
P9	38	36	12	31	9	7	20	39	14	34
P10	16	5	40	34	10	8	15	23	26	37

Tabel XI menunjukkan frekuensi kemunculan bilangan hasil acak. Dari sampel indeks yang ditandai, terlihat dengan jelas bahwa frekuensi kemunculan bilangan yang sama cukup banyak, seperti angka 10 muncul hingga 6 kali, angka 4 yang muncul hingga 4 kali, angka 23 yang muncul hingga lima kali, angka 18 yang muncul tiga kali, dan seterusnya. Hal ini disebabkan Fisher-Yates hanya dapat mengacak hingga 40 bilangan.

Angka yang ditandai dengan warna pada Tabel XI dan Tabel XII hanyalah sebagian karena hanya digunakan untuk menunjukkan bahwa banyak indeks angka yang muncul kembali.

Apa yang tampak pada Tabel XI dan Tabel XII menunjukkan bahwa walaupun besar range indeks adalah 100 tidak berpengaruh terhadap nilai indeks yang berulang, namun memiliki pengaruh terhadap banyaknya indeks dengan angka yang sama berulang.



Gambar 8. Frekuensi Kemunculan Indeks pada Uji Pertama

Pada Gambar 8 dapat dilihat bahwa kemunculan indeks paling banyak dua kali, dengan beberapa yang muncul hingga tiga-empat kali. Adapun yang tampil pada grafik merupakan gabungan dari indeks yang sudah ada. Berdasarkan hasil perhitungan tersebut, angka indeks terbanyak yang muncul adalah angka 10 dan 36, masing-masing bernilai 6 kali dengan 9 kali angka indeks yang tidak berulang.

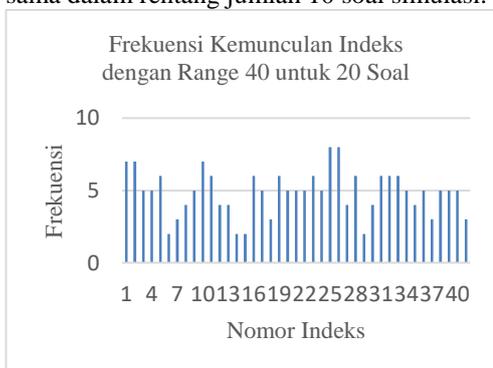
Dari 100 kali pengacakan indeks (10 pengguna dengan masing-masing 10 soal), hanya 9 kali index yang muncul sekali. Dengan demikian, penggunaan indeks 40 untuk pengambilan 10 soal dapat dikatakan tidak dapat memenuhi perhitungan indeks yang baik.

Tabel XII.

Hasil Pengujian Fisher-Yates Ketiga – Uji Kemunculan Indeks yang Sama pada No Soal yang Berbeda Soal No-

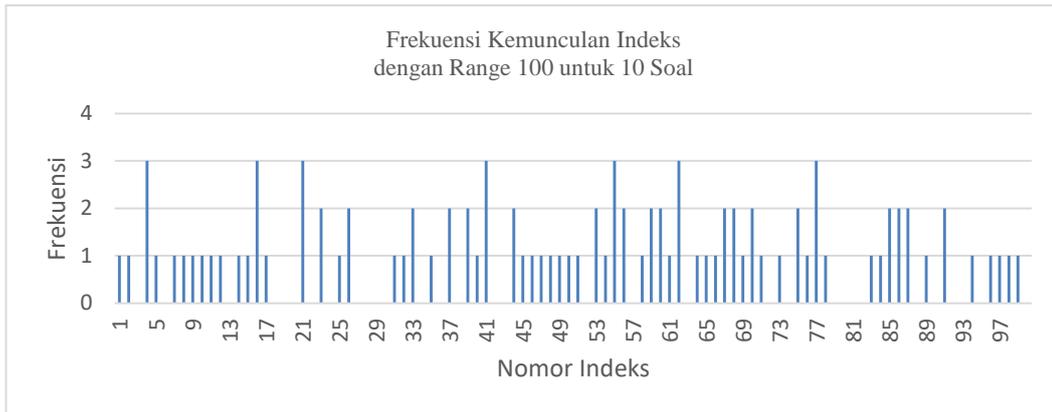
Peserta	1	2	3	4	5	6	7	8	9	10
P1	15	59	62	37	84	61	87	41	39	16
P2	41	53	56	96	60	26	70	65	68	86
P3	17	76	83	4	39	35	58	55	77	21
P4	54	14	1	4	48	50	32	16	91	7
P5	45	31	66	67	9	75	5	46	68	10
P6	69	77	44	11	21	37	98	40	85	97
P7	47	51	49	60	62	94	23	56	78	99
P8	25	55	33	23	75	12	86	26	59	89
P9	4	16	8	21	87	91	33	70	73	64
P10	62	67	55	44	53	77	85	71	41	2

Pada Tabel XII terlihat bahwa banyaknya perulangan indeks yang berulang. Namun, berbeda dengan yang terjadi pada Tabel XI, tabel ini sebagian besar hasil indeks bernilai tunggal dan indeks yang hanya berulang sekali, dengan kata lain tidak ada indeks yang muncul lebih dari dua kali. Panjang bilangan acak sebesar 100 indeks pada uji coba ini dapat mengurangi frekuensi penggunaan indeks yang sama dalam rentang jumlah 10 soal simulasi.



Gambar 9. Frekuensi Kemunculan Indeks pada Uji Ketiga

Sedikit berbeda dengan hasil uji pertama, Gambar 9 menunjukkan bahwa banyak di antara indeks tersebut yang muncul lebih dari dua kali. Nilai indeks terbanyak yang muncul dari hasil uji ketiga ini adalah 25 dan 26, masing-masing bernilai 8 kali dengan tidak ada indeks yang tidak berulang. Angka indeks yang muncul paling sedikit dua kali terjadi pada angka 6, 14, 15, dan 29. Oleh karena sering munculnya angka berulang, maka jumlah bank soal sebanyak 40 soal dapat dikatakan tidak cocok digunakan untuk pengambilan 20 soal karena dari 100 kali pengacakan soal, tidak ada satu pun angka yang muncul hanya sekali.

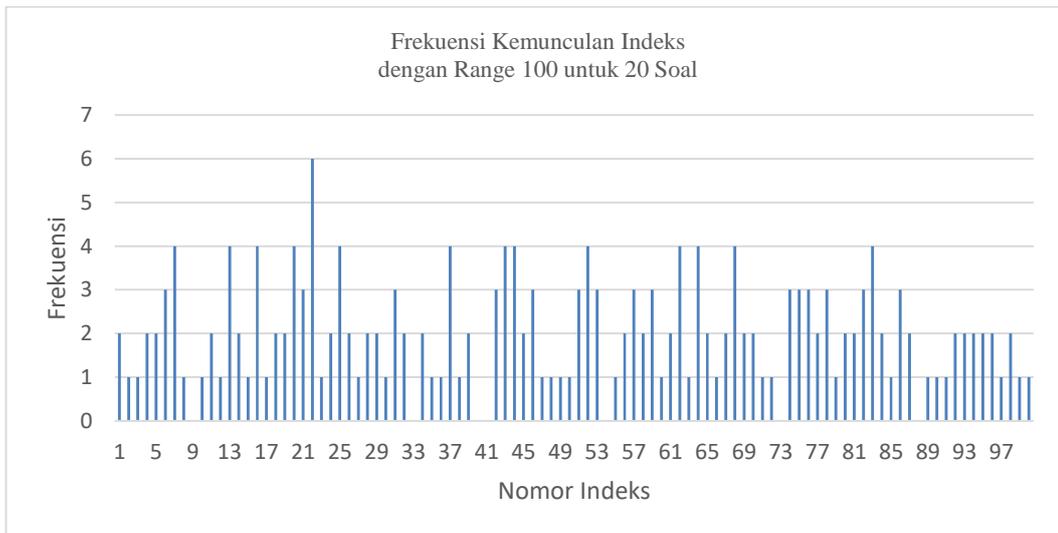


Gambar 10. Frekuensi Kemunculan Indeks pada Uji Kedua

Apabila Tabel VIII diolah lebih lanjut, maka dapat diperoleh angka statistika frekuensi kemunculan indeks yang dapat dilihat pada Gambar 10.

Berdasarkan Gambar 10, nilai tertinggi pengulangan kemunculan indeks adalah tiga kali, yang terjadi pada indeks nomor 4, 16, 41, 55, 62, dan 77. Selain itu terdapat pula 31 nomor indeks yang tidak muncul sama sekali. Kemunculan indeks berulang pada pengacakan 100 indeks soal pada Gambar 10 menunjukkan

bahwa dengan 100 bank soal perkiraan munculnya satu nilai indeks adalah tiga kali. Dengan demikian, untuk pengambilan 10 soal dari 100 soal bank soal, pengacakan ini dapat dikatakan jauh lebih baik daripada hasil uji pertama pada Gambar 8. Akan tetapi, masih banyak nilai indeks soal yang berulang setidaknya satu kali, yang berdasarkan Gambar 10 adalah 26 kali dari 100 kali pengocokan indeks.



Gambar 11. Frekuensi Kemunculan Indeks pada Uji Keempat

Gambar 11 menunjukkan bahwa meskipun ada nomor yang tidak muncul sama sekali karena berada di luar banyak soal yang diambil, tetapi kemunculan indeks berulang mengalami peningkatan yang signifikan jika dibandingkan dengan yang terlihat pada Gambar 10. Hal ini membuktikan bahwa banyak soal yang akan dimunculkan kepada pengguna mempengaruhi pengulangan indeks.

Berdasarkan hasil penelitian tersebut, dapat dikatakan pula bahwa perulangan indeks dapat

dipengaruhi oleh jumlah soal dalam bank data. Untuk mengurangi terjadinya pengulangan penggunaan indeks yang sama, jumlah soal yang ada di dalam bank soal setidaknya 10 kali banyaknya soal yang akan diambil.

IV. KESIMPULAN

Pembangkit bilangan acak pada algoritma Fisher-Yates merupakan salah satu penyebab munculnya kembali indeks yang sama. Oleh

sebab itu, kemunculan indeks berulang adalah relatif

Jika dirunut berdasarkan algoritma Fisher-Yates, semua indeks yang berada di dalam *range* pasti akan muncul, sehingga perulangan indeks pada sisi pengguna pasti terjadi. Akan tetapi, dengan dibatasinya jumlah soal yang akan dikeluarkan kepada pengguna dan memperbesar *range* dapat menekan kemungkinan terjadinya pengulangan tersebut.

Maka berdasarkan hasil penelitian ini, seperti yang telah dijelaskan sebelumnya, dapat diambil kesimpulan bahwa semakin banyak soal yang akan ditampilkan kepada pengguna, semakin besar pula kemungkinan munculnya indeks berulang.

REFERENSI

- [1] Ade Ibijola, Abejide Olu, *A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures*, [Online]. Available: <https://pdfs.semanticscholar.org/d9f6/65a955a2706c7dd3d4fe47e959f0d118932b.pdf>, 2012
- [2] Derek O'Connor, *A Historical Note on Shuffle Algorithm*, [Online]. Available: https://www.academia.edu/1205620/OConnor_-_A_Historical_Note_on_Shuffle_Algorithms, 2014
- [3] Ryan Nugraha, *Penerapan Algoritma Fisher-Yates pada Aplikasi the Lost Insect untuk Pengenalan Jenis Serangga berbasis Unity 3D*, STMIK Global Informatika, Palembang, 2015.
- [4] Ekojono, *Penerapan Algoritma Fisher-Yates pada Pengacakan Game Soal Aritmetika*, Politeknik Negeri, Malang, 2017.
- [5] Fandi Ahmad Rizal, Budi Suyanto, Tri Raharjo Yudiantoro, *Aplikasi Game Edukasi Matematika dengan Konsep Aritmatika Anak Berbasis Android*, Jurnal Teknik Elektro Terapan Politeknik Negeri Semarang, 2016.
- [6] Nur Aisyah, *Implementasi Metode Fisher-Yates Shuffle untuk Pengacakan Pertanyaan pada Game Ali and the Labirin*, UIN Maulana Malik Ibrahim, Malang, 2016.
- [7] Akbar Gani dan Linda Marlina, *Aplikasi Pembelajaran Trigonometri Berbasis Android menggunakan Algoritma Fisher-Yates Shuffle*, SMTIK Nusa Mandiri, Jakarta, 2017.
- [8] Beki Subaeki dan Dicky Ardiansyah, *Implementasi Algoritma Fisher-Yates Shuffle pada Aplikasi Multimedia Interaktif untuk Pembelajaran Tenses Bahasa Inggris*, Universitas Sangga Buana YPKP, Bandung, 2017.
- [9] Yanuarida Tri Cahyono, *Rancang Bangun Aplikasi Tes IQ Berbasis Web dengan Pengacakan Soal Menggunakan Algoritma Fisher-Yates Shuffle*, Universitas Jember, Jember, 2015.
- [10] Linda Marlinda, *Sistem Basis Data*, Penerbit Andi, Yogyakarta, 2004

Adapun, banyaknya data pada bank soal yang diperlukan agar mengurangi kemungkinan terjadinya pengulangan nilai indeks adalah sepuluh kali banyaknya soal yang akan diujikan kepada peserta simulasi.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada STMIK AKAKOM yang turut memberikan dukungan baik secara moril maupun materil untuk mempublikasikan karya ini. Terima kasih juga kepada Kepala Litbang dan PPM STMIK AKAKOM yang turut mendorong untuk mempublikasikan karya ini.