



IMPLEMENTASI METODE *LINEAR CONGRUENTIAL GENERATOR* (LCG) PADA TES IQ BERBASIS ANDROID

Ramadhin Zaid^{*1}, Sutardi², Bambang Pramono³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari

e-mail: ^{*1}ramadhinzaid@gmail.com, ²sutardi_ft@uho.ac.id, ³ bambangpramono09@gmail.com

Abstrak

Secara umum, kecerdasan mengacu pada kemampuan bawaan seseorang untuk mempelajari berbagai hal, memikirkan solusi terhadap sebuah masalah, dan kemampuan menyesuaikan diri dengan situasi tertentu. Tes kecerdasan atau tes *Intelligence Quotient* (IQ) digunakan secara luas, kemudian dikembangkan dalam berbagai tes keterampilan dan tes bakat lainnya. Metode *Linear Congruential Generator* (LCG) atau pembangkit blangan acak kongruen-lanjar merupakan pembangkit bilangan acak yang sederhana, mudah dimengerti teorinya, dan juga mudah untuk diimplementasikan. Keunggulan LCG terletak pada kecepatannya dan hanya membutuhkan sedikit operasi bit. Dengan menerapkan Metode *Linear Congruential Generator* (LCG) dalam melakukan Tes IQ berbasis android, berdasarkan hasil pengujian *Chi-Square* soal teracak sesuai dengan keinginan.

Kata kunci; LCG, Tes IQ, *Chi-Square*

Abstract

In general, intelligence refers to a person's innate ability to learn things, think of solutions to a problem, and the ability to adapt to certain situations. Intelligence tests or Intelligence Quotient (IQ) tests are widely used, then developed in various skills tests and other aptitude tests. The Linear Congruential Generator (LCG) method is a random number generator that is simple, easy to understand, and also easy to implement. The advantage of LCG lies in its speed and only requires a few bits of operations. By applying the Linear Congruential Generator (LCG) Method in conducting an Android-based IQ Test, based on the Chi-Square test results randomized questions as desired.

Keywords; LCG, IQ Test, *Chi-Square*

1. PENDAHULUAN

Secara umum, kecerdasan mengacu pada kemampuan bawaan seseorang untuk mempelajari berbagai hal, memikirkan solusi terhadap sebuah masalah, dan kemampuan menyesuaikan diri dengan situasi tertentu. Di sisi lain, tes *Intelligence Quotient* (IQ) adalah sebuah cara sistematis untuk mengukur tingkat kecerdasan seseorang dengan memberikan pertanyaan dan masalah yang telah disusun sedemikian rupa. Sangat penting untuk digaris bawahi bahwa

kecerdasan bukanlah mengenai seberapa banyak seseorang memiliki pengetahuan dan keahlian, tetapi lebih ke seberapa cepat seseorang mampu menyerap suatu pengetahuan dan keahlian baru baginya.

Dalam buku *Intelligence: Knowing and Measuring*, Profesor Resing dan Drenth menjawab pertanyaan apa inteligensi itu sebenarnya. Mereka mendefinisikan inteligensi sebagai "seluruh kemampuan kognitif dan intelektual yang diperlukan untuk memperoleh pengetahuan, dan menggunakan pengetahuan itu dalam cara yang tepat untuk



memecahkan suatu masalah yang memiliki tujuan jelas dan terstruktur”.

Istilah *intelligence quotient* atau IQ kali pertama diperkenalkan pada awal abad ke-20 oleh seorang psikolog Jerman bernama William Stern. Sejak itu, tes kecerdasan atau tes *Intelligence Quotient* (IQ) digunakan secara luas, kemudian dikembangkan dalam berbagai tes keterampilan dan tes bakat lainnya. Namun, hal ini kembali memicu perdebatan dan kontroversi terkait dengan penggunaan tes kecerdasan, bias budaya, pengaruh kecerdasan bahkan hingga cara kita mendefinisikan kecerdasan itu sendiri [1].

Mengacu pada buku Hitung Sendiri IQ Anda, terdapat 450 soal untuk tes IQ dimana dalam satu sesi diberikan 30 soal dengan waktu 60 menit. Dalam aplikasi android memungkinkan untuk pengguna dapat masuk dalam permainan lebih dari satu kali. Oleh karena itu, diperlukan sebuah algoritma untuk mengacak soal pada saat pengguna masuk kedalam permainan tes IQ agar pengguna tidak menemukan soal yang sama saat melakukan tes IQ.

Berkaitan dengan itu, Metode *Linear Congruential Generator* (LCG) atau jika diubah kedalam bahasa Indonesia menjadi Pembangkit Bilangan Acak Kongruen-Lanjar merupakan pembangkit bilangan acak yang sederhana, mudah dimengerti teorinya, dan juga mudah untuk diimplementasikan. Keunggulan LCG terletak pada kecepatannya dan hanya membutuhkan sedikit operasi bit. Sayangnya, LCG tidak dapat digunakan untuk kriptografi karena bilangan acaknya dapat diprediksi urutan kemunculannya. Oleh karena itu LCG tidak aman digunakan untuk kriptografi. Namun demikian, LCG tetap berguna untuk aplikasi non-kriptografi seperti simulasi, sebab LCG memperlihatkan sifat statistik yang bagus dan sangat tepat untuk uji-uji empirik [2].

Android adalah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android merupakan sistem operasi yang terbuka atau *open source*. Dengan sumber kode yang terbuka memungkinkan setiap perangkat lunak bebas untuk memodifikasi dan disebar. Dengan pengembangan sumber kode terbuka, Android berkembang pesat mengalahkan sistem operasi yang lain. Android diawali dengan rilis beta

pada tahun 2007. Versi komersial yaitu Android 1.0 pada september 2008. Penamaan versi Android sesuai dengan urutan abjad. Urutan menurut level *Application Programming Interface* (API) yang digunakan dalam pengembangan sistem operasi menghubungkan antar muka perangkat keras dan sistem operasi [3].

Fungsi dan peran perangkat *smartphone* android tidak dapat dipungkiri sangat membantu dalam media pembelajaran. Cara ini sangatlah tepat mengingat pengguna *smartphone* android sudah beranjak tidak hanya untuk orang dewasa tapi bahkan anak kecil sudah banyak menggunakannya. Tercatat pada <https://gs.statcounter.com/> pengguna *smartphone* android di Indonesia pada tahun 2015 mencapai 68.75% [4].

2. METODE PENELITIAN

2.1 Teori Kecerdasan

Dalam buku Hitung Sendiri IQ Anda menjelaskan beberapa teori utama kecerdasan yang muncul selama 100 tahun terakhir.

a. Charles Spearman – Kecerdasan Umum (1863-1945)

Kecerdasan terdiri atas kemampuan umum yang dinamakan dengan faktor-g (*general factor*) dan kemampuan khusus yang dinamakan dengan faktor-s (*special factor*). Ia berpendapat bahwa faktor-g dimiliki oleh semua individu, seperti kemampuan untuk menyelesaikan masalah, sedangkan faktor-s dipengaruhi oleh pengalaman (lingkungan), seperti kemampuan untuk memainkan alat musik

b. Louis L. Thurstone – Kemampuan Mental Primer (1887-1955)

Teori Thurstone berfokus pada tujuh “kemampuan mental primer” yang berbeda. Kemampuan yang ia gambarkan adalah pemahaman *verbal* (*verbal comprehension*), pemikiran/penalaran (*reasoning*), kecepatan perseptual (*perceptual speed*), kemampuan numerik (*numerical ability*), kelancaran kata (*word fluency*), memori *asosiatif*/daya ingat (*memory*), dan visualisasi spasial (*spatial factor*).

c. Howard Gardner – Kecerdasan Ganda

Teori Howard Gardner menjelaskan delapan kecerdasan berbeda yang didasarkan pada keterampilan dan kemampuan yang dihargai dalam budaya yang berbeda. Delapan kecerdasan tersebut adalah kecerdasan *visual-spasial*, kecerdasan *verbal-linguistik*, kecerdasan kinestetik-*jasmani*, kecerdasan logis-matematis, kecerdasan *interpersonal*, kecerdasan musikal, kecerdasan *intrapersonal*, dan kecerdasan naturalistik.

d. Robert Sternberg – Teori Kecerdasan Triarki

Inteligensi atau kecerdasan adalah kemampuan yang berhubungan dengan seberapa baik individu berkaitan dengan perubahan lingkungan sepanjang umur mereka dengan tiga faktor yang berbeda yaitu kecerdasan analitis, kecerdasan kreatif dan kecerdasan praktis.

2.2 Klasifikasi *Intelligence Quotient* (IQ)

Pada umumnya, rentang skor tes IQ yang banyak dilakukan saat ini merupakan perpaduan tiga metode dari tes Stanford-Binet, Lewis Terman, dan Wechsler, yang didapatkan seperti yang ditunjukkan pada Tabel 1 berikut ini.

Tabel 1 Klasifikasi IQ

Skor	Klasifikasi
70-79	Tingkatan IQ rendah (keterbelakangan mental)
80-89	Tingkatan IQ rendah yang masih dalam kategori normal (bodoh)
91-110	Tingkatan IQ normal (rata-rata)
111-120	Tingkatan IQ tinggi dalam kategori normal (cerdas)
121-130	Tingkatan IQ <i>superior</i> (sangat cerdas)
>131	Tingkatan IQ sangat <i>superior</i> (<i>genius</i>)

2.3 *Linear Congruential Generator* (LCG)

Linear Congruential Generator atau jika diubah kedalam bahasa Indonesia menjadi Pembangkit Bilangan Acak Kongruen-Lanjar merupakan pembangkit bilangan acak yang sederhana, mudah dimengerti teorinya, dan juga mudah untuk diimplementasikan [2].

Linear Congruential Generator didefinisikan dalam relasi berulang seperti yang ditunjukkan pada Persamaan (1) berikut.

$$X_n = (aX_{n-1} + b) \text{ mod } m \quad (1)$$

Keterangan:

- X_n = Bilangan acak ke-n dari deretnya
- X_{n-1} = Bilangan acak sebelumnya
- a = Faktor pengali
- b = *Increment*
- m = *Modulus*

X_0 adalah kunci pembangkit atau disebut juga umpan (*seed*). *Linear Congruential Generator* (LCG) mempunyai periode tidak lebih besar dari m, dan pada kebanyakan kasus periodenya kurang dari itu. LCG mempunyai periode penuh (m – 1) jika memenuhi syarat berikut:

- a. b relatif prima terhadap m
- b. a – 1 dapat dibagi dengan semua faktor prima dari m dan a – 1 adalah kelipatan 4 jika m adalah kelipatan 4
- c. m > maks (a, b, X_0)
- d. a > 0, b > 0

Contoh membangkitkan bilangan acak sebanyak 10 kali dengan a=1, b=19, m=30 dan $X_0=2$:

1. $X_n = (a(X_{n-1}) + b) \text{ mod } m$
 $X_1 = (1 * 1 + 19) \text{ mod } 30$
 $= 20 \text{ mod } 30$
 $= 20$
2. $X_n = (a(X_{n-1}) + b) \text{ mod } m$
 $X_2 = (1 * 20 + 19) \text{ mod } 30$
 $= 39 \text{ mod } 30$
 $= 9$
3. $X_n = (a(X_{n-1}) + b) \text{ mod } m$
 $X_3 = (1 * 9 + 19) \text{ mod } 30$
 $= 28 \text{ mod } 30$
 $= 19$
4. $X_n = (a(X_{n-1}) + b) \text{ mod } m$
 $X_4 = (1 * 19 + 19) \text{ mod } 30$
 $= 28 \text{ mod } 30$
 $= 28$
5. $X_n = (a(X_{n-1}) + b) \text{ mod } m$
 $X_5 = (1 * 28 + 19) \text{ mod } 30$
 $= 47 \text{ mod } 30$
 $= 17$
6. $X_n = (a(X_{n-1}) + b) \text{ mod } m$
 $X_6 = (1 * 17 + 19) \text{ mod } 30$
 $= 36 \text{ mod } 30$
 $= 6$

7. $X_n = (a(X_{n-1}) + b) \bmod m$
 $X_7 = (1 * 6 + 19) \bmod 30$
 $= 25 \bmod 30$
 $= 25$
8. $X_n = (a(X_{n-1}) + b) \bmod m$
 $X_8 = (1 * 25 + 19) \bmod 30$
 $= 44 \bmod 30$
 $= 14$
9. $X_n = (a(X_{n-1}) + b) \bmod m$
 $X_9 = (1 * 14 + 19) \bmod 30$
 $= 33 \bmod 30$
 $= 3$
10. $X_n = (a(X_{n-1}) + b) \bmod m$
 $X_{10} = (1 * 3 + 19) \bmod 30$
 $= 22 \bmod 30$
 $= 22$

Maka bilangan acak yang dibangkitkan adalah: 20, 9, 19, 28, 17, 6, 25, 14, 3, 22. Untuk iterasi perhitungan bilangan acak yang dibangkitkan dapat dilihat pada Tabel 2 berikut ini.

Tabel 2 Contoh iterasi LCG

Iterasi	X0	M	a	b	Xi
1	1	30	1	19	20
2	20	30	1	19	9
3	9	30	1	19	19
4	19	30	1	19	28
5	28	30	1	19	17
6	17	30	1	19	6
7	6	30	1	19	25
8	25	30	1	19	14
9	14	30	1	19	3
10	3	30	1	19	22

2.3 Metode Chi-Square

Chi-Square test atau tes kai kuadrat tergolong kedalam jenis statistik nonparametrik sehingga *Chi-Square test* tidak memerlukan syarat data berdistribusi normal. *Chi Kuadrat* dapat digunakan untuk menguji hipotesis deskriptif satu sampel atau satu variabel, yang terdiri atas dua kategori atau lebih. selain itu dapat digunakan untuk menguji hipotesis komparatif 2 sampel atau 2 variabel serta untuk menguji hipotesis asosiatif yang berskala nominal [5]

Uji *Chi-Square* dapat dirumuskan seperti ditunjukkan pada Persamaan (2).

$$x^2 = \sum \frac{(f_o - f_h)^2}{f_h} \quad (2)$$

Keterangan:

$x^2 = Chi-Square$

$f_o =$ Frekuensi bilangan acak

$f_h =$ Frekuensi yang diharapkan

2.4 Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android merupakan sistem operasi yang terbuka atau *open source*. Dengan sumber kode yang terbuka memungkinkan setiap perangkat lunak bebas untuk memodifikasi dan disebar. Dengan pengembangan sumber kode terbuka, android berkembang pesat mengalahkan sistem operasi yang lain.

Android diawali dengan rilis beta pada tahun 2007. Versi komersial yaitu Android 1.0 pada september 2008. Penamaan versi Android sesuai dengan urutan abjad. Urutan menurut level *Application Programming Interface* (API) yang digunakan dalam pengembangan sistem operasi menghubungkan antar muka perangkat keras dan sistem operasi [2].

2.5 Flutter

Flutter adalah sebuah *Software Development Kit* (SDK) atau *framework open-source* yang dikembangkan oleh Google untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi Android dan iOS. Flutter menggunakan bahasa pemrograman Dart dalam pengkodean. Perbedaan *framework* Flutter dengan yang lainnya yaitu dalam *build* aplikasi, pada *framework* ini semua kodenya di *compile* dalam kode *native*-nya (Android NDK, LLVM, AOTcompiled) tanpa ada *interpreter* pada prosesnya sehingga proses *compile*-nya menjadi lebih cepat [6].

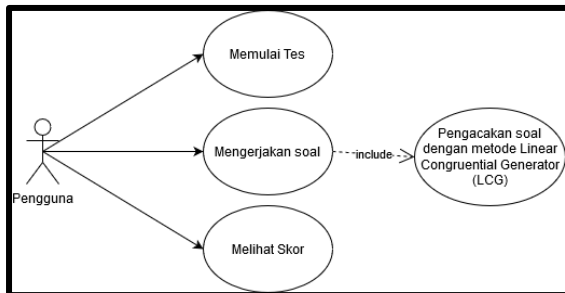
3. HASIL DAN PEMBAHASAN

Pembuatan program dan hasil uji coba program yang telah dirancang kemudian diimplementasikan dengan bahasa pemrograman dart. Uji coba dilakukan untuk mengetahui apakah program dapat berjalan sebagaimana mestinya dengan lingkungan uji

coba yang telah ditentukan serta dilakukan sesuai dengan skenario uji coba.

3.1 Use Case Diagram

Use case diagram digunakan untuk memodelkan fungsionalitas sistem yang dilihat dari pengguna yang ada di luar sistem (aktor). *Use case diagram* dapat dilihat pada Gambar 1.



Gambar 1 Use case diagram

3.2 Implementasi Sistem

Implementasi tampilan terbagi atas beberapa bagian utama yaitu:

1) Tampilan *Home*

Tampilan *Home* merupakan tampilan utama pada aplikasi tes IQ berbasis Android. Implementasi dari desain tampilan *home* menampilkan logo tes IQ, sepatah kata mengenai tes IQ, instruksi mengenai tombol *button* untuk memulai tes IQ, dan *button* Mulai untuk memulai mengerjakan beberapa soal dalam tes IQ sesuai dengan yang ditunjukkan pada Gambar 2.



Gambar 2 Tampilan *Home*

2) Tampilan Soal

Tampilan Soal merupakan halaman yang ditampilkan setelah menekan *button*

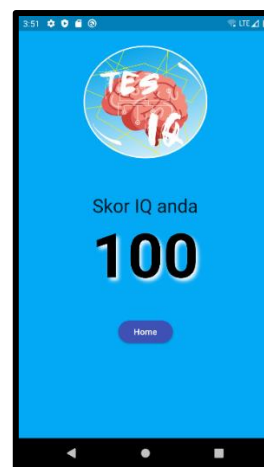
mulai pada halaman *home*. Pada halaman ini, terdapat *appbar* yang berisi logo tes IQ, waktu mengerjakan soal dan *button* untuk langsung melihat hasil atau skor IQ yang didapatkan, kemudian setelah *appbar* terdapat bagian yang menampilkan soal baik itu berupa gambar maupun teks, kemudian bagian yang menampilkan 5 pilihan jawaban yang dapat dipilih dengan menekan *button* seperti yang ditunjukkan pada Gambar 3.



Gambar 3 Tampilan Soal

3) Tampilan Skor

Tampilan Skor merupakan halaman yang ditampilkan ketika pengguna telah selesai dalam mengerjakan soal-soal pada halaman soal atau ketika pengguna menekan *button* lihat hasil. Pada halaman ini menampilkan logo tes IQ, skor IQ, dan *button* *home* untuk kembali di halaman *home* seperti yang ditunjukkan pada Gambar 4.



Gambar 4 Tampilan Skor

3.2 Pengujian *Chi-Square*

Pengujian ini dilakukan untuk memeriksa hasil pengacakan soal untuk Tes IQ yang telah dikelompokkan dalam 5 kelompok probabilitas. Pengujian ini dilakukan kepada 5 *user* dengan cara *user* menggunakan aplikasi Tes IQ dan melihat kombinasi soal yang tampil.

a) Pengujian *Chi-Square* pertama

Pengujian ini dilakukan untuk memeriksa hasil pengacakan paket soal untuk Tes IQ yang telah dikelompokkan dalam 5 kelompok probabilitas. Pengujian ini dilakukan dengan cara *user* menggunakan aplikasi Tes IQ dan melihat kombinasi soal yang tampil. Hasil pengujian ditunjukkan pada Tabel 3 berikut:

Tabel 3 Pengujian *Chi-Square* pertama

Soal	Bilangan Teracak	Fo	Fh	$(F_o - F_h)^2 / F_h$
1-20	20, 15, 10, 5, 19, 7, 6, 14	8	6	0.67
21-40	39, 34, 29, 38, 33	5	6	0.167
41-60	58, 53, 48, 43, 57, 52	6	6	0
61-80	77, 72, 67, 62, 71	5	6	0.167
81-100	96, 91, 86, 81, 100, 95	6	6	0
Jumlah		30	30	1

Pengujian pertama:

1. Dalam perhitungan ternyata diperoleh *Chi* kuadrat hitung = 1, selanjutnya harga ini dibandingkan dengan harga *Chi* kuadrat tabel dengan dk (derajat kebebasan) $5 - 1 = 4$.
2. Berdasarkan tabel *Chi* kuadrat, bahwa bila dk = 9 dan kesalahan = 10 %, harga *Chi* kuadrat = 7,779
3. Karena nilai *Chi* kuadrat dari hasil perhitungan data yang dikerjakan diperoleh 1, artinya mempunyai nilai lebih kecil dari harga *Chi* kuadrat tabel (7,779), maka dapat disimpulkan (H_0) diterima atau pendistribusian paket soal pemain tersebut dinyatakan normal atau dapat terdistribusi sesuai hasil yang diinginkan.

b) Pengujian *Chi-Square* kedua

Pengujian ini dilakukan untuk memeriksa hasil pengacakan paket soal untuk Tes IQ yang telah dikelompokkan dalam 5 kelompok probabilitas. Pengujian ini

dilakukan dengan cara *user* menggunakan aplikasi Tes IQ dan melihat kombinasi soal yang tampil. Hasil pengujian ditunjukkan pada Tabel 4 berikut:

Tabel 4 Pengujian *Chi-Square* kedua

Soal	Bilangan Teracak	Fo	Fh	$(F_o - F_h)^2 / F_h$
1-20	16, 11, 6, 1, 20, 15	6	6	0
21-40	21, 35, 30, 25, 39, 34, 40	7	6	0.167
41-60	59, 54, 49, 44, 58, 53	6	6	0
61-80	78, 73, 68, 63, 77, 72	6	6	0
81-100	97, 92, 87, 82, 96	5	6	0.167
Jumlah		30	30	0.33

Pengujian kedua:

1. Dalam perhitungan ternyata diperoleh *Chi* kuadrat hitung = 1, selanjutnya harga ini dibandingkan dengan harga *Chi* kuadrat tabel dengan dk (derajat kebebasan) $5 - 1 = 4$.
2. Berdasarkan tabel *Chi* kuadrat, bahwa bila dk = 9 dan kesalahan = 10 %, harga *Chi* kuadrat = 7,779
3. Karena nilai *Chi* kuadrat dari hasil perhitungan data yang dikerjakan diperoleh 0.33, artinya mempunyai nilai lebih kecil dari harga *Chi* kuadrat tabel (7,779), maka dapat disimpulkan (H_0) diterima atau pendistribusian paket soal pemain tersebut dinyatakan normal atau dapat terdistribusi sesuai hasil yang diinginkan.

c) Pengujian *Chi-Square* ketiga

Pengujian ini dilakukan untuk memeriksa hasil pengacakan paket soal untuk Tes IQ yang telah dikelompokkan dalam 5 kelompok probabilitas. Hasil pengujian ditunjukkan pada Tabel 5 berikut.

Tabel 5 Pengujian *Chi-Square* ketiga

Soal	Bilangan Teracak	Fo	Fh	$(F_o - F_h)^2 / F_h$
1-20	17, 12, 7, 2, 16	5	6	0.167
21-40	22, 36, 31, 26, 40, 21, 35	7	6	0.167
41-60	41, 60, 55, 50, 45, 59, 54	7	6	0.167
61-80	79, 74, 69, 64, 78, 73	6	6	0

81-100	98, 93, 88, 83, 97	5	6	0.167
Jumlah		30	30	0.67

Pengujian ketiga:

1. Dalam perhitungan ternyata diperoleh *Chi* kuadrat hitung = 1, selanjutnya harga ini dibandingkan dengan harga *Chi* kuadrat tabel dengan dk (derajat kebebasan) $5 - 1 = 4$.
2. Berdasarkan tabel *Chi* kuadrat, bahwa bila dk = 9 dan kesalahan = 10 %, harga *Chi* kuadrat = 7,779
3. Karena nilai *Chi* kuadrat dari hasil perhitungan data yang dikerjakan diperoleh 0.67, artinya mempunyai nilai lebih kecil dari harga *Chi* kuadrat tabel (7,779), maka dapat disimpulkan (H_0) diterima atau pendistribusian paket soal pemain tersebut dinyatakan normal atau dapat terdistribusi sesuai hasil yang diinginkan.

d) Pengujian *Chi-Square* keempat

Pengujian ini dilakukan untuk memeriksa hasil pengacakan paket soal untuk Tes IQ yang telah dikelompokkan dalam 5 kelompok probabilitas. Hasil pengujian ditunjukkan pada Tabel 6 berikut.

Tabel 6 Pengujian *Chi-Square* keempat

Soal	Bilangan Teracak	Fo	Fh	(Fo-Fh) ² /Fh
1-20	18, 13, 8, 3, 17	5	6	0.167
21-40	23, 37, 32, 27, 22, 36	6	6	0
41-60	42, 56, 51, 46, 41, 60, 55	7	6	0.167
61-80	61, 80, 75, 70, 65, 79, 74	7	6	0.167
81-100	99, 94, 89, 84, 98	5	6	0.167
Jumlah		30	30	0.67

Pengujian keempat:

1. Dalam perhitungan ternyata diperoleh *Chi* kuadrat hitung = 1, selanjutnya harga ini dibandingkan dengan harga *Chi* kuadrat tabel dengan dk (derajat kebebasan) $5 - 1 = 4$.
2. Berdasarkan tabel *Chi* kuadrat, bahwa bila dk = 9 dan kesalahan = 10 %, harga *Chi* kuadrat = 7,779
3. Karena nilai *Chi* kuadrat dari hasil perhitungan data yang dikerjakan diperoleh 0.67, artinya mempunyai nilai

lebih kecil dari harga *Chi* kuadrat tabel (7,779), maka dapat disimpulkan (H_0) diterima dan dinyatakan normal atau dapat terdistribusi sesuai hasil yang diinginkan.

e) Pengujian *Chi-Square* kelima

Pengujian ini dilakukan untuk memeriksa hasil pengacakan paket soal untuk Tes IQ yang telah dikelompokkan dalam 5 kelompok probabilitas. Hasil pengujian ditunjukkan pada Tabel 7 berikut.

Tabel 7 Pengujian *Chi-Square* kelima

Soal	Bilangan Teracak	Fo	Fh	(Fo-Fh) ² /Fh
1-20	19, 14, 9, 4, 18	5	6	0.167
21-40	24, 38, 33, 28, 23, 37	6	6	0
41-60	43, 57, 52, 47, 42, 56	6	6	0
61-80	62, 76, 71, 66, 61, 80, 75	7	6	0.167
81-100	81, 100, 95, 90, 85, 99	6	6	0
Jumlah		30	30	0.33

Pengujian kelima:

1. Dalam perhitungan ternyata diperoleh *Chi* kuadrat hitung = 1, selanjutnya harga ini dibandingkan dengan harga *Chi* kuadrat tabel dengan dk (derajat kebebasan) $5 - 1 = 4$.
2. Berdasarkan tabel *Chi* kuadrat, bahwa bila dk = 9 dan kesalahan = 10 %, harga *Chi* kuadrat = 7,779
3. Karena nilai *Chi* kuadrat dari hasil perhitungan data yang dikerjakan diperoleh 0.33, artinya mempunyai nilai lebih kecil dari harga *Chi* kuadrat tabel (7,779), maka dapat disimpulkan (H_0) diterima dan dinyatakan normal atau dapat terdistribusi sesuai hasil yang diinginkan.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai implementasi *Linear Congruential Generator* (LCG), maka diperoleh kesimpulan sebagai berikut:

1. Metode *linear congruential generator* (LCG) dapat diterapkan dalam pengacakan soal yang akan ditampilkan saat melakukan Tes IQ di *Smartphone* Android dengan kecepatan pengoperasian dan

- keunikan setiap paket soal yang muncul ketika memulai aplikasi.
2. Dari pengujian *Chi-Square* yang dilakukan untuk menguji keacakan Metode *Linear Congruential Generator* (LCG). Dari 5 kali pengujian mendapatkan hasil uji yaitu H_0 diterima. Artinya, soal pada tes IQ berbasis android teracak sesuai keinginan.

5. SARAN

Beberapa saran yang perlu diperhatikan untuk pengembangan penelitian selanjutnya adalah aplikasi ini tentunya dapat dikembangkan lagi dengan menambahkan bidang tes IQ yang diperlukan.

DAFTAR PUSTAKA

- [1] Tim Smart Solution, *Hitung Sendiri IQ Anda*. Yogyakarta: B First, 2015.
 - [2] H. E. Putra and K. Harianto, "Implementasi Linear Congruential Generator untuk Pengacakan Gambar Pada Permainan Puzzle," *SATIN - Sains dan Teknologi. Inf.*, vol. 4, no. 1, p. 89, 2018.
 - [3] K. Ramadhan, L. W. Astuti, and D. A. Verano, "Game Edukasi Tebak Gambar Bendera Negara Menggunakan Metode Linear Congruential Generator (LCG) Berbasis Android," *J. Inform. Glob.*, vol. Vol.6, no. 1, pp. 27–32, 2015.
 - [4] G. E. Pribadi, U. Syaripudin, and W. Uriawan, "Aplikasi Pembelajaran Bahasa Sunda Dengan Implementasi Algoritma Linear Congruential Generator Dan Fuzzy Berbasis Android," *J. Online Inform.*, vol. 1, no. 1, p. 34, 2016.
 - [5] Y. N. Sufren, *Mahir Menggunakan SPSS Secara Otodidak*. Jakarta: PT. Alex Media Komputindo, 2013.
 - [6] F. E. Krisnanda and R. Tanone, "Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis Mobile menggunakan Flutter," vol. 5, pp. 281–295, 2019.
-