

Rancang Bangun *Game IQ Test Preparation* dengan Mengimplementasikan Metode *FSM*

Muhammad Maheswara Raditya^{#1}, Fauziah^{#2}, Ratih Titi Komalasari^{#3}

[#]Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional

Jl. Sawo Manila, Pasar Minggu Jakarta 12520 Indonesia

¹raditya1026@gmail.com

²fauziah@civitas.unas.ac.id

³ratih.titi@unas.civitas.ac.id

Abstrak— Pada umumnya tes IQ akan menggunakan kertas sebagai medianya, namun sebenarnya hal tersebut dinilai kurang efisien sebab perkembangan teknologi dapat digunakan untuk meminimalisir penggunaan kertas. Metode Finite State Machine (FSM) sesuai dalam penerapan alur sistem dan Construct 2 merupakan tools yang sesuai dalam merancang game quiz. Penelitian ini menggunakan metode FSM sebagai metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan state (keadaan), event (kejadian) dan action (aksi). Dalam state machine sistem menanggapi satu state (keadaan) dan Construct 2 sebagai tools perancangan aplikasi. Diagram implementasi metode FSM pada penelitian ini dimulai dari q_0 hingga q_n , adalah state awal dari dimulainya game tes IQ ini. Pengujian dilakukan pada 8 tipe Android yang dimulai dari versi android 4.0 Ice Cream Sandwich hingga android versi 9.0 Pie. Perhitungan pada aplikasi juga telah dibandingkan dengan perhitungan secara manual untuk mengetahui seberapa besar tingkat akurasi yang didapatkan. Hasil dari pengujian yang dilakukan sebanyak 70 kali terhadap 3 range usia didapatkan margin error sebesar 0,45%, maka didapat tingkat akurasi sebesar 99,55%.

Kata kunci— Android, Construct 2, Finite State Machine, Game, Tes IQ

I. PENDAHULUAN

Pada umumnya tes *IQ* akan menggunakan kertas sebagai medianya, namun sebenarnya hal tersebut dinilai kurang efisien sebab perkembangan teknologi dapat digunakan untuk meminimalisir penggunaan kertas. Maka pembuatan aplikasi tes *IQ* akan lebih efisien tidak hanya untuk lingkungan tetapi juga untuk user karena dapat memudahkan *user* karena aplikasi ini bersifat mobile.

Pada penelitian yang dilakukan oleh Nyimas Sopiha dan Usman Ependi menghasilkan aplikasi tes *IQ* berbasis Android yang disertakan dengan hasil test [1]. Penelitian yang dilakukan oleh Rahmat Tullah menghasilkan aplikasi tes *IQ* berbasis web dengan rincian perhitungan soal [2].

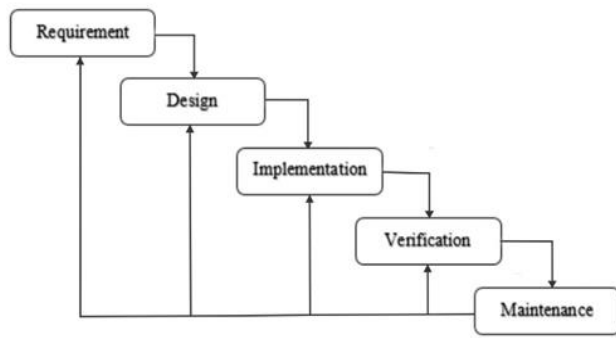
Penelitian yang dilakukan oleh Affis Masturina Nisa dan Hafidz Kurniansyah menerapkan Finite State Machine dalam perancangan game berbasis desktop dan menghasilkan game yang telah sesuai dengan state yang dirancang [3].

Dari hasil penelitian yang dilakukan sebelumnya terdapat dua metode yang bisa digunakan, yaitu metode FSM dan *Forward Chaining*. Konsep FSM apabila diberikan suatu input maka state akan berpindah ke *state* selanjutnya hingga *state* akhir, konsep FSM sesuai dengan game seperti kuis. Konsep *Forward Chaining* serupa dengan FSM, namun perpindahan *state* dilakukan berdasarkan bagaimana *input state* sebelumnya dipilih, konsep *Forward Chaining* sesuai dengan konsep game seperti interaktif game [4]. Penelitian sebelumnya dapat digabungkan menjadi penelitian yang peneliti telah lakukan, maka peneliti melakukan pembaharuan dari penelitian terdahulu dengan menggunakan metode *finite state machine* diimplementasikan pada perancangan game *IQ Test Preparation* dengan menggunakan Construct 2. Dilihat dari penelitian terdahulu maka metode *finite state machine* merupakan metode yang sesuai dengan aplikasi yang dibangun pada penelitian ini, penelitian ini dibuat bertujuan untuk membatasi penggunaan kertas dan memanfaatkan teknologi yang ada, selain itu dengan adanya aplikasi ini dapat memudahkan *user* karena aplikasi ini bersifat *mobile*.

II. METODE PENELITIAN

A. Waterfall Model

Pada penelitian ini digunakan metode waterfall model yang digambarkan pada Gambar 1.



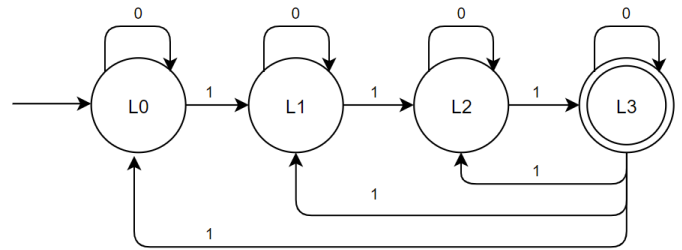
Gambar. 1 Waterfall model

Tahapan yang dilakukan pada penelitian ini terdiri dari:

- 1) *Requirement* (Kebutuhan)
 Pada tahap ini, dibutuhkan spesifikasi perangkat yang digunakan untuk merancang dan menguji aplikasi.
- 2) *Design* (Desain)
 Pada tahap ini akan mendesain aplikasi yang dirancang sebelum nantinya dilakukan tahapan implementasi dengan pemrograman.
- 3) *Implementation* (Implementasi)
 Setelah didapat desain yang diinginkan, kemudian disesuaikan desain dengan pemrograman untuk menyesuaikan dengan desain yang telah dirancang.
- 4) *Verification* (Verifikasi)
 Tahap verifikasi diperlukan untuk melakukan pengujian pada aplikasi yang telah dirancang, dari hasil verifikasi menyatakan keberhasilan atau terdapat kesalahan.
- 5) *Maintenance* (Pemeliharaan)
 Pada tahap maintenance diperlukan untuk memelihara aplikasi jika terjadi error dikemudian hari atau ada penambahan fitur.

B. Finite State Machine

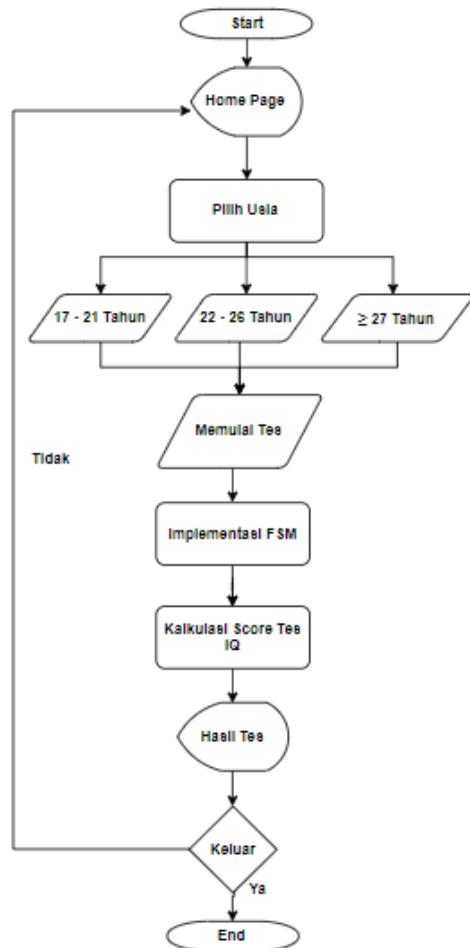
Finite State Machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan *state* (keadaan), *event* (kejadian) dan *action* (aksi). Dalam *state machine* sistem menempati satu *state* (keadaan) [5]. Sistem akan beralih atau bertransisi menuju ke *state* lain jika mendapatkan masukan *event* tertentu. Sistem akan tetap melakukan aksi yang sama pada suatu *state* sampai sistem menerima *event* tertentu baik yang berasal dari perangkat luar atau komponen dari sistem itu sendiri [6]. Setiap *state* terhubung oleh transisi dan setiap transisinya mengarah ke *state* lainnya. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi sederhana yang melibatkan rangkaian proses yang relatif rumit [7].



Gambar. 2 Finite state machine

Diagram pada Gambar 2 menunjukkan alur dari FSM, dimulai dari state awal, apabila diberikan input nilai 0 maka akan kembali ke state tersebut, dan apabila diberikan input nilai 1, maka akan lanjut ke state berikutnya hingga mencapai state akhir, dan pada state akhir dapat kembali ke state – state berikutnya yang sudah dilewati [8]. Maka aplikasi ini akan dibuat dengan mengimplementasikan konsep dari FSM yang mana pada setiap perpindahan soal akan berpindah pada state selanjutnya lalu tahap akhir atau state penerima akan menampilkan hasil score dari tes IQ yang telah dikerjakan [5][6].

Game dibuat dengan rancangan *flowchart* yang dapat dilihat pada Gambar 3.

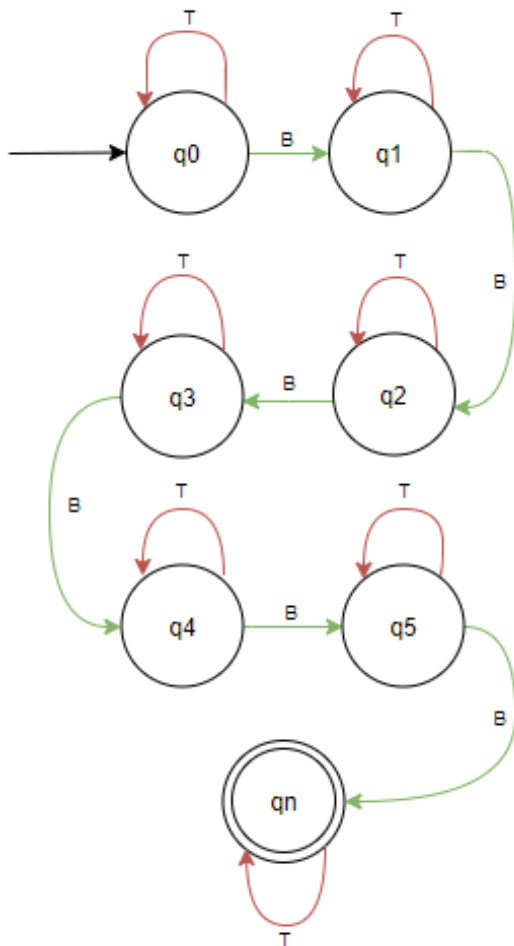


Gambar. 3 Flowchart rancangan aplikasi

Pada *flowchart* diatas menjelaskn bahwa alur game dimulai dari :

1. Tampilan *Home Page*
2. Kemudian, dilakukan *input* untuk menampilkan menu pilih usia,
3. Lalu dilakukan *input* kembali untuk memulai *test*,
4. FSM diimplementasikan pada proses saat perpindahan soal satu ke soal berikutnya, dari implemetasi tersebut, *user* tidak dapat kembali pada soal *IQ Test* yang telah dijawab, sehingga akan langsung diarahkan sesuai *state* yang telah dirancang,
5. Lalu hasil *test* dikalkulasi secara otomatis untuk mendapatkan *score IQ*,
6. Setelah kalkulasi selesai, maka *score* akan ditampilkan,
7. Setelah itu *user* dapat memilih untuk keluar dari aplikasi atau kembali ke *home page*.

Implementasi metode penelitian pada *game*:



Gambar. 4 Diagram *Finite State Machine* pada tes IQ

Pada Gambar 4 merupakan diagram implementasi metode *Finite State Machine* pada penelitian ini. Dimulai dari q0, adalah *state* awal dari dimulainya game tes IQ ini. Keterangan pada gambar dapat dilihat pada Tabel 1 [9].

TABEL I
TABEL KETERANGAN DIAGRAM STATE

State	Keterangan
(q0,q1,... qn)	Soal IQ Test
B	Berikutnya
T	Diam

Kemudian akan dilakukan penilaian dengan *range score* pada tes IQ, yang dapat dilihat pada Tabel 2 dibawah ini [10].

TABEL II
PENILAIAN

Nomor Soal	Score
1 - 10	25
11 - 30	30
31 - 35	30
36 - 60	45

Pada Tabel 2 menerangkan perihal *score* penilaian pada tiap soal, dari soal nomor 1 – 10 merupakan jenis soal *figural*, nomor 11-30 merupakan jenis soal *verbal*, nomor 31-35 merupakan jenis soal *numerikal*, nomor 36-60 merupakan jenis soal *perseptual*. Jumlah keseluruhan dari total soal yang yang digunakan merupakan sample dari IQ Test yang sesungguhnya. Apabila dijawab benar semua, maka akan didapat *total score* sebesar 25 dan seterusnya hingga soal nomor 60. Dari penilaian tersebut akan dilakukan penggabungan *total score* [11][12], dengan rumus sebagai berikut :

$$IQ = \frac{MA}{CA} \times 100 \tag{1}$$

Dimana:

MA = *Mental Age*

CA = *Chronological Age*

IQ = *Intelligence Quotient*

Setelahnya akan diukur indeks IQ dengan mengukur pada Tabel 3.

TABEL III
INDEKS SCORE DAN PENILAIAN IQ TEST

Score	Indeks
< 70	Gangguan Kognitif
70 - 89	Kecerdasan Normal Rendah
90 - 110	Kecerdasan Normal Sedang
111 - 120	Kecerdasan Normal Tinggi
121 - 130	Sangat Cerdas / Superior
> 130	Genius

Tabel 3 mendeskripsikan detail penilaian pada *game* yang dirancang, dimulai dari *score* lebih kecil dari 70 yang artinya *user* memiliki gangguan kognitif, hingga *score* diatas 130 yang artinya *user* memiliki tingkat kecerdasan yang sangat tinggi atau jenius [12].

Rumus perhitungan konversi dari *score* angka ke persentase untuk menentukan *error* [13]. Rumus (2)

digunakan untuk menghitung hasil dari perhitungan otomatis menggunakan aplikasi.

$$(nA - x) \times 100 = \text{hasil} \div 180 = n \quad (2)$$

Rumus (3) digunakan dengan perhitungan secara manual sebagai pembandingan validasi dari perhitungan otomatis.

$$(nM - y) \times 100 = \text{hasil} \div 180 = n \quad (3)$$

Dari hasil rumus (2) dan (3) dihitung menggunakan rumus (4) dimana akan dihasilkan persentase *error* yang ada.

$$\text{error} = n1 - n2 \quad (4)$$

Dimana :

- nA = Score perhitungan aplikasi
- nM = Score perhitungan manual
- x = 60
- y = 240
- Pembagi = $y - x = 180$

Contoh perhitungan membandingkan error pada aplikasi dilakukan sebagai berikut:

$$(126 - 60) = 66 * 100 = 6600 / 180 = 36,6\%$$

$$(125 - 60) = 65 * 100 = 6500 / 180 = 36,1\%$$

$$\text{Error} = 36,6\% - 36,1\% = \mathbf{0,5\%}$$

III. HASIL DAN PEMBAHASAN

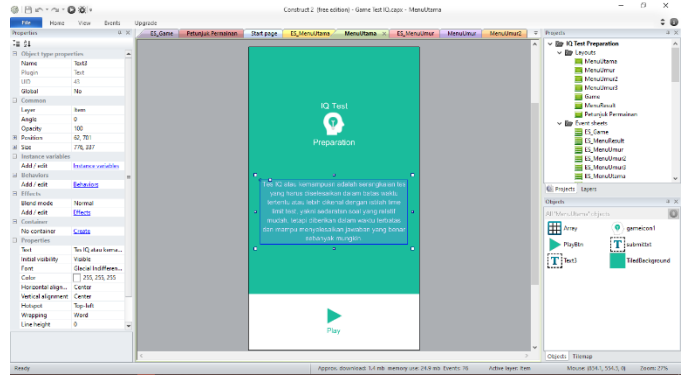
A. Analisa Kebutuhan Sistem

Pada perancangan aplikasi *game IQ test*, dibutuhkan perangkat pendukung berupa perangkat keras (*hardware*) maupun perangkat lunak (*software*), yang terdiri dari:

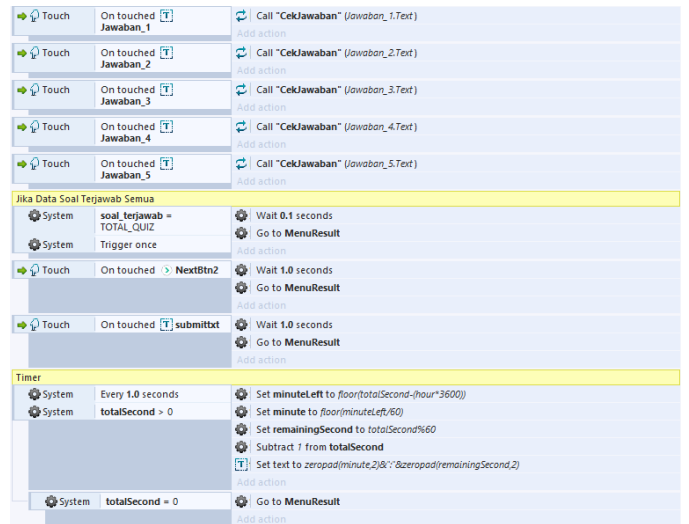
1. Kebutuhan *Hardware*: Komputer Desktop
Intel® Core™ i7-8700 3.20 GHz 64bit dengan RAM 16GB dan AMD X4 950 3.50GHz 64bit dengan RAM 8GB.
2. Kebutuhan *Software*:
 - a. Draw.io, digunakan membuat *flowchart*.
 - b. Figma, digunakan untuk membuat *user interface* dari aplikasi yang dibangun.
 - c. Adobe PhoneGap, digunakan untuk mengubah hasil export HTML5 dari Construct 2 ke APK.
 - d. Construct 2, digunakan untuk membuat dan membangun *game IQ Test Preparation* yang penulis teliti.
 - e. Photoshop CC 2017, digunakan untuk mengedit soal bergambar pada tes *IQ*.

B. Penggunaan Tools

Gambar 5 menunjukkan halaman untuk mengatur layout pada aplikasi yang peneliti bangun, dimulai dari ukuran layar hingga assets – assets yang digunakan dalam aplikasi [14].



Gambar 5. Pembuatan Layout pada Construct 2



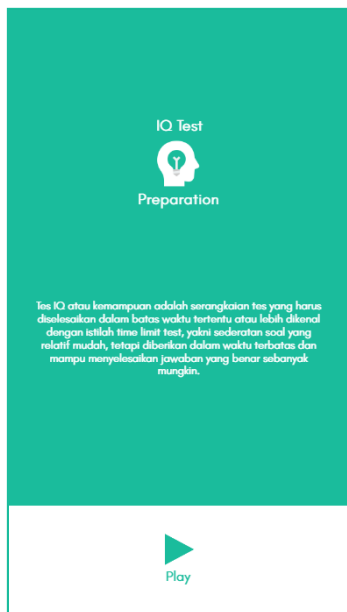
Gambar 6. Pengaturan aplikasi

Gambar 6 menampilkan pengaturan yang digunakan untuk menyesuaikan alur aplikasi pada tes *IQ* yang dibangun agar aplikasi berjalan sesuai *logic* dari FSM yang dirancang sebelumnya [15].

C. Tampilan Aplikasi

Dengan merancang user interface maka sesuai dengan penelitian ini sebab penelitian ini bertujuan untuk mengurangi penggunaan kertas dan memaksimalkan penggunaan teknologi berbasis Android.

Penelitian ini dirancang dengan membangun *User Interface* menggunakan Figma dan kemudian dimasukkan lagi kedalam Construct 2. Tampilan yang dibangun dapat dilihat pada gambar berikut [16].



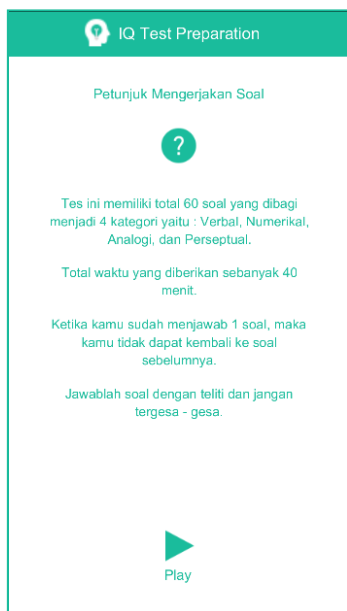
Gambar. 7 User interface menu utama

Gambar 7 merupakan tampilan *menu* utama pada saat *game* dimulai. *Menu* utama hanya menampilkan informasi mengenai tes *IQ* dan *button play* untuk memulai *quiz*.



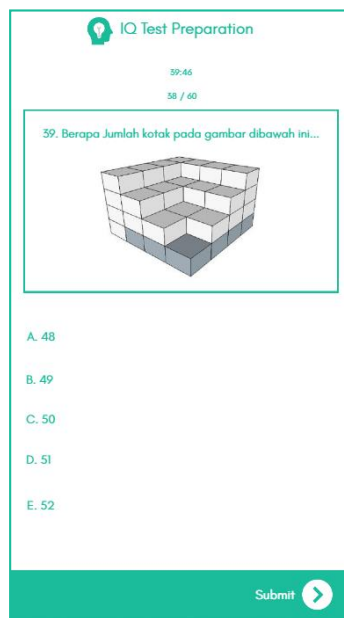
Gambar. 9 User interface Pilih Usia

Gambar 9 merupakan halaman pilih usia, tampilan halaman ini akan terbuka setelah menekan tombol *play* pada gambar sebelumnya.



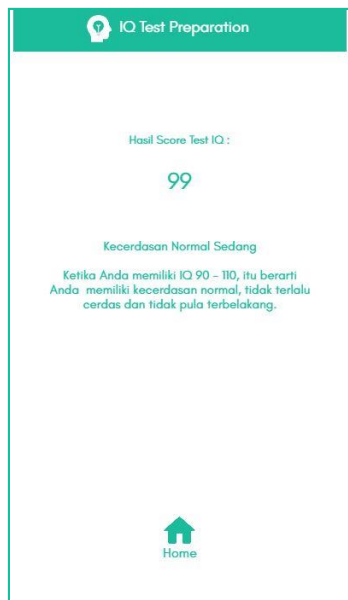
Gambar. 8 User interface petunjuk soal

Gambar 8 menampilkan *layout* petunjuk cara bermain dan dijelaskan kategori soal yang ada hingga waktu yang diberikan untuk mengisi soal – soal tes *IQ*



Gambar. 10 User interface test IQ

Gambar 10 akan menampilkan soal – soal tes yang akan *user* kerjakan, terdiri dari 60 soal dan diberi waktu sebanyak 40 menit. soal terdiri dari 4 kategori soal, yaitu analogi, verbal, numerikal, dan perseptual.



Gambar. 11 User interface hasil score

Gambar 11 akan menampilkan hasil *score* dari tes *IQ* beserta indeks dari *score* yang didapat dari soal – soal yang sudah dikerjakan.

D. Pengujian

Pengujian pada *device* Android dilakukan bertujuan untuk mengetahui aplikasi yang dirancang telah *responsive* dan *Compatible*. Indikator keberhasilan dalam pengujian berdasarkan versi Android dan resolusi layar. Pengujian dilakukan pada 8 tipe Android yang memiliki spesifikasi masing – masing yang berbeda dan pengujian juga dimulai dari versi Android 4.0 *Ice Cream Sandwich* hingga Android versi 9.0 *Pie*.

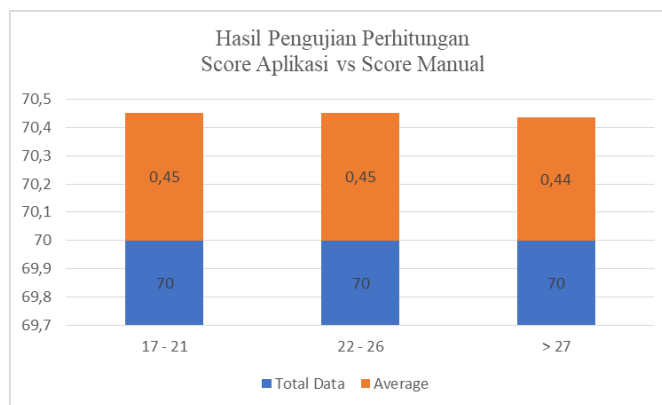
Berikut merupakan tabel pengujian pada *smartphone* yang memiliki versi android 4 hingga 9:

TABEL IV
PENGUJIAN SMARTPHONE DAN VERSI ANDROID

Jenis Device	Keterangan
Android 1 1. OS Android 4.0.3 (<i>Ice Crean Sandwich</i>) 2. Resolusi Layar 480 x 800 pixels	Berhasil dijalankan
Android 2 1. OS Android 4.2.1 (<i>Jellybean</i>) 2. Resolusi Layar 720 x 1280 pixels	Berhasil dijalankan
Android 3 1. OS Android 4.4.2 (<i>Kitkat</i>) 2. Resolusi Layar 540 x 960 pixels	Berhasil dijalankan
Android 4 1. OS Android 5.0.2 (<i>Lollipop</i>) 2. Resolusi Layar 720 x 1280 pixels	Berhasil dijalankan
Android 5 1. OS Android 6.0.1 (<i>Marshmallow</i>) 2. Resolusi Layar 720 x 1280 pixels	Berhasil dijalankan
Android 6 1. OS Android 7.0.0 (<i>Nougat</i>) 2. Resolusi Layar 1080 x 1920 pixels	Berhasil dijalankan

Android 7 1. OS Android 8.1.0 (<i>Oreo</i>) 2. Resolusi Layar 1080 x 2160 pixels	Berhasil dijalankan
Android 8 1. OS Android 9.0 (<i>Pie</i>) 2. Resolusi Layar 1080 x 2340 pixels	Berhasil dijalankan

Pada Tabel 4 merupakan tabel pengujian pada 8 tipe *smartphone* yang memiliki versi Android berbeda - beda dimulai dari android 4.0 *Ice Cream Sandwich* hingga yang terbaru versi android 9.0 *Pie*, dan pula didapatkan hasil pengujian bahwa game *IQ Test Preparation* dapat dijalankan dibeberapa *smartphone* yang memiliki RAM yang berkapasitas mulai dari 512 MB hingga 4GB [4].



Gambar 12. Hasil Perhitungan *score* aplikasi vs manual

Pada Gambar 12 ditampilkan tabel *charts* dari hasil pengujian perhitungan hasil *score* tes *IQ* yang dihitung secara manual dibandingkan dengan perhitungan melalui aplikasi. Dari usia 17-21 didapatkan *error* sebesar 0,45%, dari usia 22-26 didapatkan *error* sebesar 0,45%, Di usia >27 didapatkan *error* sebesar 0,45%, Dari 3 range usia yang sudah ditentukan peneliti melakukan 70 kali pengujian kepada 3 range usia tersebut. Lalu didapatkan *margin error* sebesar 0,45%, maka dari itu peneliti mendapatkan hasil tingkat keakurasian sebesar 99,55%.

IV. KESIMPULAN

Berdasarkan hasil yang diperoleh oleh penulis, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi Game *IQ Test Preparation* dapat digunakan untuk mengurangi penggunaan kertas dan dalam prosesnya user dapat mengerjakan dimanapun sebab lebih efisien karena dapat di *install* dalam Android.
2. Hasil dari pengujian didapatkan aplikasi ini dapat dijalankan di Android yang memiliki spesifikasi minimal v.4.0.3 dan juga telah dilakukan perhitungan manual untuk membandingkan perhitungan melalui aplikasi dengan tingkat akurasi sebesar 99,55%.
3. Aplikasi dibangun dengan menggunakan metode FSM sebab memiliki konsep yang sesuai dengan aplikasi game yang dirancang.

REFERENSI

- [1] N. Sopiah and U. Ependi, "Aplikasi Latihan Tes Iq Menggunakan Android," *J. Ilm. Matrik*, no. 3, pp. 163–172, 2017.
- [2] R. Tullah, M. Iqbal, and A. S. Cahyadi Putra, "Perancangan Aplikasi Simulasi Tes IQ Untuk Tingkat SLTA Berbasis Web," vol. 5, no. 1, pp. 3–8, 2015.
- [3] H. K. Affis Masturina Nisa, "Perancangan dan Implementasi Finite State Automata pada Pusheen Cat Maze Game dengan Adobe Flash," vol. 2, no. 1, pp. 13–18, 2019.
- [4] A. Irmawati, "Perancangan Aplikasi Tes IQ Siswa untuk Pertimbangan Pemilihan Jurusan dengan Metode Forward Chaining," vol. 3, no. 2354–5771, pp. 102–112, 2016.
- [5] R. A. Ma'arif and F. Fauziah, "Implementasi Finite State Automata (FSA) dalam Proses Pengisian Kartu Rencana Studi," *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 3, no. 3, pp. 115–120, 2018.
- [6] T. I. Saputra and A. Gunaryati, "Simulasi Vending Machine Dengan Mengimplementasikan Finite State Automata," vol. 3, no. 3, pp. 143–148, 2018.
- [7] M. F. Rahadian, A. Suyatno, and S. Maharani, "Penerapan Metode Finite State Machine Pada Game 'The Relationship'," vol. 11, no. 1, pp. 14–22, 2016.
- [8] S. Rostianingsih, S. B. Gregorius, and H. K. Wijaya, "Game Simulasi Finite State Machine Untuk Pertanian Dan Peternakan," *J. DKV Adiwarna*, vol. 5, pp. 2–7, 2016.
- [9] A. Ahyuna and I. Irmawati, "Perancangan Aplikasi Tes IQ Siswa untuk Pertimbangan Pemilihan Jurusan dengan Metode Forward Chaining," *Creat. Inf. Technol. J.*, vol. 3, no. 2, p. 102, 2016.
- [10] V. Maarif, A. E. Widodo, and D. Y. Wibowo, "Aplikasi Tes IQ Berbasis Android," *ijse.bsi.ac.id IJSE – Indones. J. Softw. Eng. ISSN*, vol. 3, no. 2, pp. 2461–690, 2017.
- [11] Purwanto, "Intelegensi : Konsep dan Pengukurannya Tes IQ," *Pendidik. dan Kebud.*, vol. 16, pp. 477–485, 2018.
- [12] I. A. Saputra, J. A. Widians, and Rosmasari, "Aplikasi Sistem Pakar Skoring Tes Iq Dengan Alat Cfit Berbasis Desktop," *Univ. Mulawarman*, vol. 2, no. 1, pp. 146–151, 2017.
- [13] FunMath, "Percentage Error," *mathsisfun*.
- [14] Y. Abbas, E. Winarno, P. Studi, T. Informatika, F. T. Informasi, and U. Stikubank, "Perancangan Game Edukasi Pengenalan Angka Dalam Bahasa," No. 2012, Pp. 347–352, 2018.
- [15] M. A. I. J. A. Y. Herry sujaini, "Rancang Bangun Aplikasi Game Matematika Menggunakan Construct 2," *J. Sist. dan Teknol. Inf.*, vol. Vol. 1, no. 1, pp. 1–5, 2016.
- [16] S. Y. Riska and W. A. Rahayu, "Perancangan Game IDO untuk Pembelajaran Kosa Kata Bahasa Inggris Menggunakan Construct 2," vol. 2, no. 1, pp. 20–36, 2018.