

**IMPLEMENTASI METODE LINEAR CONGRUENTIAL  
GENERATOR (LCG) PADA PERANCANGAN GAME  
EDUKASI “SOLVE THE MATH” SEBAGAI MEDIA  
PEMBELAJARAN ANAK USIA DINI BERBASIS ANDROID**

SKRIPSI



Oleh:

ARDIYANSAH

181011400481

**PROGRAM STUDI  
TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS  
PAMULANG  
2022**

**IMPLEMENTASI METODE LINEAR CONGRUENTIAL  
GENERATOR (LCG) PADA PERANCANGAN GAME  
EDUKASI “SOLVE THE MATH” SEBAGAI MEDIA  
PEMBELAJARAN ANAK USIA DINI BERBASIS ANDROID**

**SKRIPSI**

Diajukan Untuk Melengkapi Salah Satu

Syarat

Memperoleh Gelar Sarjana Komputer



Oleh:

ARDIYANSAH

181011400481

**PROGRAM STUDI  
TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS  
PAMULANG  
2022**

## ABSTRAK

Seiring dengan pesatnya perkembangan teknologi di era saat ini, teknologi memiliki peranan yang sangat penting dalam setiap aspek kehidupan manusia. Perkembangan dan kebutuhan teknologi saat ini dapat dibuat menjadi praktis dan menarik di ponsel berbasis android, salah satunya adalah dalam dunia pendidikan seperti *game* edukasi yang berguna untuk meningkatkan semangat anak-anak dalam proses pembelajaran dengan media yang lebih menyenangkan. Penelitian ini bertujuan untuk memberikan kontribusi dalam dunia *game*, khususnya *game* yang bersifat edukatif. Dalam hal ini peneliti membuat sebuah alternatif pembelajaran berupa *game* edukasi yang di dalamnya terdapat pengoperasian bilangan matematika dasar. Metode pengembangan aplikasi ini menggunakan metode RAD (*Rapid Application Development*) dengan mengimplementasikan algoritma *Linear Congruential Generator* untuk pengacakan angka pada soal. Hasil yang didapatkan dari penelitian ini yaitu terselesaikannya *game* edukasi berbasis mobile “*Solve The Math*” dengan menggunakan Unity 3D. Dengan adanya *game* edukasi “*Solve The Math*” dapat menarik minat anak usia dini untuk belajar matematika dasar.

**Kata Kunci:** *Game* Edukasi, *Linear Congruential Generator*, Matematika

## **ABSTRACT**

*Along with the rapid development of technology in the current era, technology has a very important role in every aspect of human life. Current technological developments and needs can be made practical and interesting on android-based phones, one of which is in the world of education such as educational games that are useful for increasing children's enthusiasm in the learning process with more fun media. This study aims to contribute to the world of games, especially educational games. In this case, the researcher makes alternative learning in the form of an educational game in which there is the operation of basic mathematical numbers. This application development method uses the RAD (Rapid Application Development) method by implementing the Linear Congruential Generator algorithm for randomizing numbers in questions. The results obtained from this study are the completion of the mobile-based educational game "Solve The Math" using Unity 3D. The educational game "Solve The Math" can attract the interests of early childhood to learn basic mathematics.*

**Keywords:** *Educational Games, Linear Congruential Generator, Mathematics*

## DAFTAR ISI

LEMBAR JUDUL.....	ii
LEMBAR PERNYATAAN .....	ii
LEMBAR PERSETUJUAN .....	iii
LEMBAR PENGESAHAN .....	iv
ABSTRAK .....	iii
<i>ABSTRACT</i> .....	iv
KATA PENGANTAR.....	vii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABLE .....	xi

### BAB I PENDAHULUAN

1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah.....	4
1.3 Rumusan Masalah.....	4
1.4 Batasan Masalah .....	4
1.5 Tujuan Penelitian .....	4
1.6 Manfaat Penelitian .....	5
1.7 Sistematika Penulisan .....	5

### BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka.....	7
2.2 Landasan Teori.....	9
2.2.1 Pengertian Perancangan .....	10
2.2.2 Pengertian Aplikasi .....	10
2.2.3 Jenis-Jenis Aplikasi.....	10
2.2.4 Fungsi Aplikasi .....	11
2.2.5 Media Pembelajaran.....	12

2.2.6	Jenis-Jenis Media Pembelajaran .....	13
2.2.7	Game .....	13
2.2.8	Sejarah Perkembangan Game .....	13
2.2.9	Perkembangan Game .....	14
2.2.10	Klasifikasi Game.....	16
2.2.11	<i>Game</i> Edukasi .....	19
2.2.12	Matematika.....	20
2.2.13	Android .....	20
2.2.14	Kelahiran Android.....	21
2.2.15	Perkembangan Android.....	21
2.2.16	Android SDK ( <i>Software Development Kit</i> ) .....	27
2.2.17	UML ( <i>Unified Modeling Language</i> ).....	27
2.3	Kerangka Penelitian .....	29
2.4	<i>Use Case</i> Diagram .....	30
2.5	<i>Activity</i> Diagram .....	31
2.6	<i>Sequence</i> Diagram .....	32
2.7	<i>Class</i> Diagram.....	34
2.8	RAD ( <i>Rapid Application Development</i> ) .....	35
2.9	Algoritma .....	36
2.10	Metode <i>Linear Congruential Generator</i> .....	36
2.11	Aplikasi Pendukung .....	37
2.12	Android Studio.....	37
2.13	Unity 3D.....	38
2.14	Visual Studio Code .....	39
2.15	Bahasa Pemrograman.....	40
2.15.1	C# .....	40
2.16	Metode Pengujian .....	42
2.16.1	Tujuan Pengujian .....	42

2.16.2 Prinsip Pengujian .....	42
2.16.3 Pengujian Black Box.....	43

### **BAB III ANALISA DAN PERANCANGAN**

3.1 Analisa Sistem .....	45
3.1.1 Analisa Masalah.....	45
3.1.2 Analisa Kebutuhan Sistem.....	46
3.2 Perancangan Aplikasi.....	47
3.2.1 Gambaran Umum Sistem Usulan.....	47
3.2.2 Prosedur Sistem Yang Diusulkan .....	48
3.3 Perancangan Sistem .....	48
3.4 Perancangan Soal.....	49
3.5 Metode <i>Linear Congruential Generator</i> .....	49
3.5.1 <i>Flowchart</i> Metode <i>Linear Congruential Generator</i> .....	49
3.5.2 Implementasi <i>Linear Congruential Generator</i> .....	50
3.6 Perancangan Aplikasi.....	50
3.6.1 <i>Use Case</i> Diagram .....	51
3.6.2 <i>Activity</i> Diagram .....	53
3.6.3 <i>Sequence</i> Diagram.....	56
3.6.4 Class Diagram.....	59
3.6.5 Perancangan Layar.....	59
3.6.5.1 Rancangan Halaman Utama.....	59
3.6.5.2 Rancangan Halaman Tentang Kami .....	60
3.6.5.3 Rancangan Halaman Pilih Level.....	60
3.6.5.4 Rancangan Halaman Pengaturan .....	61
3.6.5.5 Rancangan Tampilan Konfirmasi Keluar .....	62

### **BAB IV IMPLEMENTASI DAN PENGUJIAN**

4.1 Implementasi.....	63
4.1.1 Spesifikasi Perangkat Keras.....	63
4.1.2 Spesifikasi Perangkat Lunak.....	64
4.1.3 Implementasi Antarmuka.....	64
4.2 Pengujian Sistem.....	69
4.2.1 Pengujian <i>Black Box</i> .....	69
4.2.2 Kasus dan Hasil Pengujian.....	71

## **BAB V PENUTUP**

5.1 Kesimpulan .....	78
5.2 Saran .....	78

<b>DAFTAR PUSTAKA .....</b>	<b>79</b>
-----------------------------	-----------



## DAFTAR GAMBAR

Gambar 2. 1 <i>Flowchart</i> Kerangka Penelitian.....	30
Gambar 2. 2 <i>Rapid Application Development</i> .....	35
Gambar 2. 3 Android Studio .....	38
Gambar 2. 4 Unity 3D .....	39
Gambar 2. 5 Visual Studio Code .....	40
Gambar 2. 6 Bahasa Pemrograman C# .....	42
Gambar 3. 1 Gambaran Umum Sistem Usulan.....	48
Gambar 3. 2 <i>Flowchart</i> Metode <i>Linear Congruential Generator</i> .....	49
Gambar 3. 3 Diagram Implementasi <i>Linear Congruential Generator</i> .....	50
Gambar 3. 4 <i>Use Case</i> Diagram.....	51
Gambar 3. 5 <i>Activity</i> Diagram Halaman Utama.....	54
Gambar 3. 6 <i>Activity</i> Diagram Tombol Bermain .....	54
Gambar 3. 7 <i>Activity</i> Diagram Tombol Pengaturan .....	55
Gambar 3. 8 <i>Activity</i> Diagram <i>About Us</i> .....	55
Gambar 3. 9 <i>Activity</i> Diagram Tombol Keluar .....	56
Gambar 3. 10 <i>Sequence</i> Diagram Halaman Utama.....	57
Gambar 3. 11 <i>Sequence</i> Diagram Bermain .....	57
Gambar 3. 12 <i>Sequence</i> Diagram Tentang Aplikasi .....	58
Gambar 3. 13 <i>Sequence</i> Diagram Keluar .....	58
Gambar 3. 14 <i>Class</i> Diagram .....	59
Gambar 3. 15 Menu Utama Aplikasi .....	60
Gambar 3. 16 Menu About Us .....	60
Gambar 3. 17 Rancangan Halaman Pilih Level .....	61
Gambar 3. 18 Rancangan Halaman Pengaturan.....	61
Gambar 3. 19 Rancangan Tampilan Konfirmasi Keluar.....	62
Gambar 4. 1 Tampilan Halaman Utama .....	65
Gambar 4. 2 Tampilan Halaman Pengaturan .....	65
Gambar 4. 3 Tampilan Halaman Keluar .....	66
Gambar 4. 4 Tampilan <i>About Us</i> .....	66
Gambar 4. 5 Tampilan Pilih Level.....	67
Gambar 4. 6 Tampilan Pertanyaan Penjumlahan .....	67

Gambar 4. 7 Tampilan Pertanyaan Pengurangan .....	68
Gambar 4. 8 Tampilan Pertanyaan Perkalian .....	68
Gambar 4. 9 Tampilan Pertanyaan Perkalian .....	69

## DAFTAR TABLE

Table 2. 1 Jenis-Jenis Diagram UML .....	28
Table 2. 2 Simbol-Simbol <i>Use Case</i> Diagram .....	30
Table 2. 3 Notasi <i>Activity</i> Diagram .....	31
Table 2. 4 Simbol <i>Sequence</i> Diagram .....	32
Table 2. 5 Simbol <i>Class</i> Diagram .....	34
Table 3. 1 Tabel <i>Use Case</i> Diagram .....	51
Table 4. 1 Perangkat Keras Yang Digunakan .....	63
Table 4. 2 Pengujian Halaman Utama.....	71
Table 4. 3 Pengujian Halaman Pertanyaan .....	73
Table 4. 4 Pengujian Tombol <i>Pause</i> .....	74
Table 4. 5 Pengujian Tombol Pengaturan .....	76

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan pesatnya perkembangan teknologi pada era waktu ini, teknologi mempunyai peranan yang sangat penting dalam setiap aspek kehidupan manusia. Segala jenis pekerjaan manusia dapat dimudahkan dengan adanya bantuan teknologi. Kebutuhan manusia pun semakin bervariasi terhadap hal yang berkaitan dengan teknologi. *Smartphone* salah satu bagian dari perkembangan teknologi saat ini. *Smartphone* memiliki berbagai macam sistem operasi dan salah satu yang paling banyak diminati masyarakat saat ini adalah android. Android merupakan sistem operasi dengan *platform* terbuka atau *open source*. Perkembangan dan kebutuhan teknologi saat ini dapat dibuat menjadi sangat praktis di ponsel berbasis Android, salah satunya adalah dalam dunia pendidikan seperti *game* edukasi yang dapat berguna untuk meningkatkan semangat anak-anak dalam proses pembelajaran dengan media yang lebih menyenangkan.

Perkembangan game begitu pesat dengan jenis yang beragam, mulai dari *game strategi*, *advanture*, *arcade*, *puzzle*, *sport*, dll yang dikemas dalam *playstation game*, *PC game* maupun *mobile device* dan akan sangat menarik bagi setiap orang terutama bagi anak-anak (Abidin, Mustain, & Wahyudi, 2017). Pada dasarnya *game* diciptakan sebagai sarana hiburan, tetapi akan lebih baik jika *game* diciptakan untuk sarana belajar supaya anak-anak bisa lebih kreatif dalam berfikir.

*Game* yang memiliki konten pendidikan lebih dikenal dengan istilah *game* edukasi. *Game* berjenis edukasi ini bertujuan untuk memancing minat belajar terhadap materi pelajaran sambil bermain, sehingga dengan perasaan senang diharapkan bisa lebih mudah memahami materi pelajaran yang disajikan. Jenis ini sebenarnya lebih mengacu kepada isi dan tujuan *game*, bukan jenis yang sesungguhnya. *Game* sangat berpotensi untuk menumbuhkan kembali motivasi belajar anak yang mengalami penurunan. Berdasarkan penelitian yang dilakukan oleh Randel pada tahun 1991 tercatat bahwa

pemakaian *game* sangat bermanfaat pada materi-materi yang berhubungan dengan matematika, fisika dan kemampuan berbahasa (seperti studi sosial, biologi dan logika).

Setiap anak pada zaman sekarang pasti pernah menggunakan *smartphone*, kebanyakan dari mereka menggunakannya untuk bermain *game*. Mereka beralasan bahwa *game* dapat dijadikan alat untuk melepas penat saat jenuh dalam belajar. Namun pada kenyataannya mereka lebih banyak memainkan *game* yang tidak bermanfaat untuk menambah ilmu pengetahuannya dan malah menjadikan ketagihan sehingga lupa waktu untuk belajar.

Masa kanak-kanak awal (usia 3 – 6 tahun) sering disebut masa pra sekolah, (Anwar, 2011). Pengertian Anak usia dini secara umum adalah anak-anak yang berusia di bawah 6 tahun. Anak pada usia 3 – 6 tahun atau usia Taman Kanak-kanak (pada jalur pendidikan formal sesuai dengan Undang-undang RI Nomor 20 tahun 2003 pasal 28 tentang Pendidikan Anak Usia Dini), merupakan masa peka bagi anak, karena masa ini merupakan masa terjadinya pematangan fungsi-fungsi fisik dan psikis yang siap merespon stimulasi lingkungan dan menginternalisasikan ke dalam pribadinya. Masa ini merupakan masa awal perkembangan fisik, kognitif, bahasa, sosial, emosional, konsep diri, disiplin, kemandirian, seni, moral, dan nilai-nilai agama. Oleh karena itu dibutuhkan suatu kondisi dan stimulasi yang sesuai dengan kebutuhan anak agar pertumbuhan dan perkembangan tercapai secara optimal (Kemendiknas, 2010).

Matematika adalah suatu pengetahuan yang mampu mengembangkan kemampuan berpikir anak. Oleh sebab itu matematika dapat sebagai sarana untuk membangun kemampuan berpikir anak mulai dari usia dini, usia pendidikan kelas awal (pendidikan dasar), pendidikan menengah, pendidikan lanjutan dan bahkan sampai mereka berada di bangku perkuliahan (Anwar, 2011)

Matematika merupakan pelajaran yang telah diberikan sejak kecil (biasanya di jenjang sekolah dasar) dan selalu ditemui dalam kehidupan sehari-hari. Beberapa hal yang dipelajari dalam operasi dasar matematika antara lain penambahan, pengurangan, perkalian, dan pembagian. Pemahaman dan penguasaan operasi dasar matematika tidak sulit dipelajari, tetapi tetap diperlukan suatu latihan khusus agar bisa melakukan perhitungan secara cepat dan benar.

Proses pengenalan bilangan pada anak usia dini memerlukan sebuah perantara yang berupa media fisik. Melalui media fisik maka anak akan mudah memahami materi yang disampaikan. Upaya pengembangan tersebut dapat dilakukan melalui kegiatan bermain seraya belajar atau belajar seraya bermain, dengan bermain anak memiliki kesempatan untuk bereksplorasi dan belajar secara menyenangkan (Depdiknas, 2003). Banyak sekali usaha yang dilakukan oleh orang tua atau pendidik pada tingkat TK untuk mengenalkan operasi bilangan. Diantaranya anak diminta untuk mengoperasikan penjumlahan atau pengurangan dengan menggunakan gambar. Dari gambar tersebut anak diminta untuk menerjemahkan kedalam bentuk angka dan mengoperasikannya. Beberapa anak mungkin berhasil menguasai keterampilan tersebut, namun banyak pula di antaranya yang masih mengalami kesulitan (Anwar, 2011).

*Game* berhitung dapat memacu seseorang untuk aktif berpikir. *Game* berhitung saat ini lebih banyak dibentuk seakan pemain sedang mengerjakan soal ulangan atau mengerjakan soal paket. Maka dari itu penulis ingin membuat suatu *game* berhitung yang interaktif sehingga anak pada usia dini akan lebih bersemangat dalam belajar matematika (Yunus, Astuti, & Khairina, 2015).

Pandemi Covid-19 yang melanda Indonesia saat ini memberikan dampak di berbagai sektor, seperti sektor kesehatan, ekonomi hingga Pendidikan (Almuttaqi, 2020). Kementerian Pendidikan dan Kebudayaan (Kemendikbud) menerbitkan Surat Edaran Nomor 15 Tahun 2020 tentang Pedoman Penyelenggaraan Belajar Dari Rumah. Dalam Masa Darurat Penyebaran Covid-19. Orangtua mengakui ketidakmampuan untuk mendampingi anak belajar di rumah. Permasalahan tersebut, membuat pembelajaran anak di rumah tidak berjalan sesuai dengan yang diharapkan. Karena harusnya pembelajaran berlangsung menyenangkan menjadi suatu kegiatan yang terpaksa harus dilakukan. Inilah yang membuat motivasi belajar anak menurun (Wahyuni, Putri, & Fadillah, 2021).

Berdasarkan uraian di atas muncul sebuah ide untuk membuat sebuah *game trivia* atau permainan pertanyaan yang dapat dimainkan oleh anak-anak dan segala kalangan untuk menyelesaikan setiap level dengan kesulitan yang berbeda, karena aman untuk anak-anak sebagai media pembelajaran. Dengan adanya *game* ini diharapkan dapat melatih anak untuk memecahkan masalah dengan caranya sendiri.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah diatas dapat diidentifikasi masalah yang ada yaitu :

1. Motivasi belajar anak mengalami penurunan.
2. Anak usia dini kesulitan dalam belajar matematika dasar.
3. Pengenalan bilangan pada anak usia dini memerlukan suatu latihan khusus sehingga memerlukan sebuah media atau perantara.

## 1.3 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah diuraikan di atas, maka dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana merancang *game* sebagai media pembelajaran yang menarik untuk anak usia dini?
2. Bagaimana mengacak angka pada soal dalam *game Solve The Math*?
3. Apakah *game* edukasi *Solve The Math* dapat menarik minat anak dalam belajar matematika?

## 1.4 Batasan Masalah

Adapun batasan masalah yang terkait dengan rumusan masalah diatas adalah sebagai berikut :

1. Isi dari *game* mengacu pada penjumlahan, pengurangan, pembagian, dan perkalian.
2. *Game* hanya bisa dijalankan pada ponsel atau tablet dengan sistem operasi berbasis android.
3. *Game* edukasi yang dibuat dikhususkan untuk anak-anak usia dini (7 tahun kebawah).
4. *Game* edukasi ini hanya dapat dimainkan oleh satu orang atau *single player*.
5. *Game* yang dibangun menggunakan grafis *2D*.

## 1.5 Tujuan Penelitian

Maksud dari tujuan peneliti ini adalah :

1. Untuk merancang sebuah game edukatif sebagai media pembelajaran yang tepat untuk anak-anak.
2. Menerapkan metode *Linear Congruential Generator* kedalam game *Solve The Math* untuk pengacakan angka pada soal.
3. Untuk mengetahui minat belajar anak-anak usia dini dalam mempelajari matematika dasar.

## 1.6 Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini adalah:

### a. Penulis

1. Sebagai jawaban atas perumusan masalah yang sudah di ajukan.
2. Memotivasi penulis untuk menciptakan karya yang lebih baik.
3. Memenuhi syarat kelulusan program pendidikan Strata-1 Jurusan Teknik Informatika di Universitas Pamulang.

### b. Universitas

1. Menambah referensi sebagai bahan penelitian lanjutan.
2. Sebagai referensi untuk membuat *game* edukasi berbasis *mobile*.

### c. Pengguna Aplikasi

1. Memberikan alternatif belajar yang menyenangkan.
2. Memberikan hiburan yang bersifat interaktif, sehingga tidak cepat merasa bosan dan jenuh.
3. Memberikan pengalaman belajar yang berbeda dan menyenangkan untuk anak-anak.

## 1.7 Sistematika Penulisan

Untuk memahami lebih jelas tentang laporan ini maka materi-materi yang tertera pada laporan ini dikelompokkan menjadi beberapa sub bab dengan sistematika penyampaian sebagai berikut:

### BAB I PENDAHULUAN

Berisi tentang latar belakang masalah, identifikasi masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian,



metode penelitian yang terdiri dari metode pengumpulan data dan metode pengembangan sistem, dan sistematika penulisan.

## **BAB II LANDASAN TEORI**

Bab ini berupa landasan teori yang membahas teori-teori dari berbagai sumber yang digunakan sebagai referensi untuk mendukung dalam penulisan penelitian ini, mengenai teori pendukung yang berhubungan dengan proses-proses perancangan aplikasi tersebut, baik teori umum maupun teori khusus.

## **BAB III ANALISA DAN PERANCANGAN**

Bab ini menjelaskan rincian atau proses dari penelitian, dimulai dari pengumpulan kebutuhan, analisis data dan perancangan sistem UML (*Unified Modeling Language*) serta rancangan perangkat sistem yang diusulkan dalam menganalisis masalah yang ada. Kemudian berusaha untuk menemukan solusi melalui sistem atau aplikasi yang sedang dikembangkan.

## **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini menjelaskan hasil implementasi dari rancangan yang telah dibuat pada bab sebelumnya, serta membahas spesifikasi khusus, *user interface* dari sistem yang diimplementasikan. Selain itu akan diuraikan hasil evaluasi dari rancangan aplikasi yang telah dibuat.

## **BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran yang berkaitan dengan pengembangan sistem, berdasarkan yang telah diuraikan pada bab-bab sebelumnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Tinjauan pustaka memuat uraian sistematis tentang hasil-hasil penelitian yang didapat oleh peneliti terdahulu yang ada hubungannya dengan penelitian yang akan dilakukan, yang diambil dari jurnal, artikel dan buku yang berhubungan dengan penelitian ini. Beberapa penelitian yang telah dilakukan adalah sebagai berikut:

- a. Pada tahun 2016, Anik Vega Vitianingsih melakukan sebuah penelitian (Jurnal INFORM Vol. 1, No. 1, (2016) ISSN: 2502-3470) dengan judul “*Game Edukasi Sebagai Media Pembelajaran Pendidikan Anak Usia Dini*” pada penelitian ini dijelaskan bahwa *game* edukasi dapat menunjang proses pendidikan. *Game* edukasi unggul dalam beberapa aspek jika dibandingkan dengan metode pembelajaran konvensional. Salah satu keunggulan yang signifikan adalah adanya animasi yang dapat meningkatkan daya ingat sehingga anak dapat menyimpan materi pelajaran dalam waktu yang lebih lama dibandingkan dengan metode pengajaran konvensional. *Game* simulasi dengan tujuan edukasi dapat digunakan sebagai salah satu media edukasi yang memiliki pola pembelajaran *learning by doing*. Berdasarkan pola yang dimiliki oleh *game* tersebut, pemain dituntut untuk belajar sehingga dapat menyelesaikan permasalahan yang ada. *Game* bisa digunakan sebagai media alternatif pembelajaran guru PAUD dalam mengubah cara belajar konvensional menjadi cara belajar simulasi *game*, sehingga dapat mengembangkan kreativitas anak, karena dalam *game* edukasi memiliki unsur tantangan, ketepatan, daya nalar dan etika.
- b. Pada tahun 2017, Benni Pane, Xaverius Najoan, Sary Paturusi melakukan sebuah penelitian (E-Jurnal Teknik Informatika Vol 12, No.1 (2017) ISSN: 2301-8364) dengan judul “Rancang Bangun Aplikasi *Game* Edukasi Ragam Budaya Indonesia” pada penelitian ini dijelaskan bahwa kemajuan teknologi membawa pengaruh bagi masyarakat, salah satunya adalah mulai menjamurnya berbagai macam *game*. Citra *game* di masyarakat masih

dipandang sebagai media yang menghibur dibanding sebagai media pembelajaran. Sifat dasar *game* yang menantang (*challenging*), membuat ketagihan (*addicted*) dan menyenangkan (*fun*) bagi mereka yang menyukai permainan, modern ini dapat berdampak negatif apabila yang dimainkan adalah *game* yang tidak bersifat edukasional. Saat ini, *game* yang beredar masih digunakan untuk hiburan dan dapat menimbulkan kecanduan. Contoh *game* yang memberikan efek kecanduan antara lain seperti; *Grand Theft Auto*, *PUBG*, *DotA*, *Point Blank*, *Call of Duty*, dll. *Game* tersebut memiliki genre *game* yang beragam. Mulai dari petualangan hingga yang paling populer adalah genre yang memiliki unsur kekerasan. *Game* dengan genre kekerasan tidak baik karena dapat menimbulkan perilaku-perilaku menyimpang. Hasil analisis menunjukkan bahwa anak yang bermain *video game* kekerasan selama beberapa tahun mencatat kenaikan tajam dalam nilai agresi mereka selama studi. Sementara, anak yang sering bermain *game* non-kekerasan tidak menunjukkan bukti peningkatan agresi. Selain membawa dampak negatif bagi generasi muda, *game* juga memiliki banyak dampak positif. Salah satunya yaitu menggunakan *game* edukasi. *Game* edukasi diharapkan dapat membantu dalam pengenalan budaya yang ada dan dapat dijadikan sebagai media pembelajaran untuk memberikan wawasan kepada anak-anak. Dengan penggunaan *game* edukasi sebagai pengenalan budaya dapat memuat materi secara tekstual, audio dan visual.

- c. Pada tahun 2017, Diana Laily Fithri dan Dave Andre Setiawan melakukan sebuah penelitian (Jurnal SIMETRIS, Vol. 8 No. 1 [April 2017] ISSN 2252-4983) dengan judul “Analisa dan Perancangan Game Edukasi Sebagai Motivasi Belajar Untuk Anak Usia Dini” pada penelitian ini dijelaskan bahwa saat ini orang tua yang sangat tinggi mobilitas kerjanya memberikan fasilitas kepada buah hatinya mulai dari usia balita hingga remaja diberi *gadget* Android untuk menghilangkan kebosanan dan orang tua mengharapkan dapat mempermudah proses pembelajaran bagi buah hati, untuk usia muda yang akan belajar membaca dan berhitung hingga usia remaja yang dapat membantu proses pembelajarannya di sekolah. Akan tetapi hal tersebut nyatanya masih sangat jauh dari harapan orang tua, yang ada di masyarakat adalah banyak buah

hati yang mendownload media sosial dan *game*, hal itu bukan menunjang pembelajaran justru menjadi menurunkan kualitas pendidikan. Sehubungan dengan hal itu dirasa penggunaan sistem dalam suatu pembelajaran sangat diperlukan, agar mengubah pembelajaran formal yang membosankan dan monoton, menjadi pembelajaran edukatif dan menyenangkan melalui perantara *game*. Oleh karena itu peneliti tertarik untuk merancang sebuah aplikasi *game* edukasi untuk anak usia dini menggunakan metode *Waterfall*. *Game* edukasi ini dirasa cocok dengan masyarakat Indonesia yang haus akan pendidikan, akan tetapi kurang meminati pembelajaran formal yang membosankan.

- d. Pada tahun 2018, Surya Amami Pramuditya., Muchammad Subali Noto dan Henry Purwono melakukan sebuah penelitian (Jurnal JNPM Vol. 2 No. 2 [September, 2018] ISSN 2549-8495) dengan judul “Desain Game Edukasi Berbasis Android Pada Materi Logika Matematika” pada penelitian ini dijelaskan dari hasil wawancara, guru telah mencoba metode pembelajaran menggunakan media *Power Point* untuk menarik minat siswa dan membantu untuk menyampaikan materi pembelajaran. Namun media *Power Point* masih belum mendukung interaksi antara siswa dan media pembelajaran, siswa juga kesulitan untuk membedakan tiap simbol atau lambang yang ada pada logika matematika. Siswa lebih tertarik untuk belajar dengan santai dan menyenangkan serta menginginkan penggunaan media pembelajaran yang menggunakan gaya belajar audio dan visual, maka dibuatlah media pembelajaran *game* edukasi. Metode yang digunakan dalam pembuatan media pembelajaran *game* edukasi ini adalah metode *Waterfall*. Hasil validasi dari *game* edukasi ini sangatlah valid (97,12%). *Game* edukasi matematika berbasis Android ini sangatlah praktis digunakan untuk semua tingkatan kemampuan siswa.

## 2.2 Landasan Teori

Landasan teori adalah seperangkat definisi, konsep serta proposisi yang telah rapi serta sistematis tentang variable-variabel dalam sebuah penelitian. Landasan teori ini akan menjadi dasar yang baik dan benar dalam sebuah penelitian menjadi

penting karena landasan teori ini menjadi sebuah pondasi serta landasan dalam penelitian tersebut (Saqid, 2015).

### **2.2.1 Pengertian Perancangan**

Perancangan adalah tahapan perancangan (desain) memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternative sistem yang terbaik (Ladjamudin, 2005).

### **2.2.2 Pengertian Aplikasi**

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga computer dapat memproses input menjadi output (Jogiyanto, 2005). Mobile diartikan sebagai perpindahan yang mudah satu tempat ke tempat lain, misalnya telepon mobile berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.

### **2.2.3 Jenis-Jenis Aplikasi**

Adapun jenis-jenis aplikasi ialah sebagai berikut:

a. Aplikasi *Desktop*

Saat ini aplikasi desktop banyak tersedia dan bisa dikategorikan ke dalam beberapa kategori. Beberapa aplikasi desktop memiliki banyak fitur.

b. Aplikasi Grafis

Adalah sebuah program untuk mengolah data yang berformat gambar baik dengan membuat gambar baru atau juga bisa mengubah gambar yang sudah dibuat sebelumnya.

c. Aplikasi *Web Browser*

Merupakan salah satu bagian penting dari internet sebagai komunitas jaringan komputer yang memberikan pelayanan http. Dengan begini, definisi teknis world wide web ialah semua sumber daya serta semua pengguna di internet yang memakai HTTP (*Hypertext Transfer Protocol*).

d. Aplikasi Multimedia

Kita semua pastinya pernah merasakan jenuh pada saat melakukan pekerjaan. Aplikasi multimedia adalah salah satu media hiburan yang dapat kita pergunakan. Apabila kita ingin mendengarkan musik MP3 atau menonton film yang sudah kita unduh, maka kita bisa memakai aplikasi ini untuk menikmatinya. Windows Media Player dan iTunes adalah aplikasi pemutar media yang populer saat ini.

e. Aplikasi *Communication*

Aplikasi ini adalah yang paling banyak digunakan dan merupakan yang paling populer. Aplikasi ini dipakai agar manusia dapat berkomunikasi dengan pengguna komputer, *smartphone* atau gadget lain. Contoh seperti aplikasi *communication* antara lain, Line, Whatsapp, BBM, dll.

f. Aplikasi DBMS

Aplikasi DBMS (*Database Management System*) dipakai untuk menghasilkan *output* berupa informasi Aplikasi seperti ini ada yang tersedia secara gratis. Tetapi ada juga yang berbayar. Contoh: *MySQL, Microsoft Access, Oracle, Foxypro*.

## 2.2.4 Fungsi Aplikasi

Berikut dibawah ini beberapa fungsi aplikasi:

a. Pendidikan

Aplikasi bisa dipakai sebagai bahan pengajaran. Misalkan saja, membuat peyajian materi dilengkapi animasi-animasi supaya lebih menarik dengan memakai *Microsoft Powerpoint*. Dengan demikian, suasana pengajaran akan memberikan kenyamanan serta mencapai hasil yang lebih maksimal.

b. Kedokteran

Tidak ada yang meyangka bahwa ada aplikasi yang dikembangkan khusus hanya untuk bidang kedokteran. Misalkan saja, adanya aplikasi untuk mendiagnosa penyakit, menawarkan perawatan rutin, bahkan sampai meracik obat.

c. Bisnis

Didalam sebuah bisnis, perlu menghitung besarnya keuntungan yang didapat. Apabila keuntungan dihitung secara manual maka akan memerlukan waktu yang lama. Oleh sebab itu diperlukannya aplikasi untuk menghitung besar keuntungan tersebut.

d. Ilmu Pengetahuan

Dengan adanya aplikasi ini akan mempermudah untuk mendapat informasi yang berkenaan dengan pengetahuan. Selain daripada itu, aplikasi ini juga bisa dipakai, apabila ingin meneliti bagian dari ilmu pengetahuan tersebut.

e. Militer

Pada zaman modern ini, banyak aplikasi yang dikembangkan khusus dalam bidang pertahanan. Misalkan saja untuk melakukan pengendalian pesawat tidak lagi menggunakan cara manual melainkan menggunakan aplikasi (Hengky, 2017).

### **2.2.5 Media Pembelajaran**

Media adalah sebuah alat yang mempunyai fungsi menyampaikan pesan. Komunikasi tidak akan berjalan tanpa bantuan sarana penyampaian pesan atau media. Media dipahami secara garis besar adalah manusia, materi, atau kejadian yang membangun kondisi yang membuat siswa mampu memperoleh pengetahuan, keterampilan, atau sikap. Pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pembelajaran merupakan bantuan yang diberikan pendidik agar dapat terjadi proses perolehan ilmu dan pengetahuan, penguasaan kemahiran dan tabiat, serta pembentukan sikap dan kepercayaan pada peserta didik. Dengan kata lain, pembelajaran adalah proses untuk membantu peserta didik agar dapat belajar dengan baik. Media pembelajaran terdiri dari beberapa jenis, masing-masing mempunyai kekurangan dan kelebihan. Dan media satu dengan media lainnya dapat saling mendukung. Media pembelajaran memiliki banyak manfaat, diantaranya untuk memperlancar interaksi antara guru dengan peserta didik sehingga pembelajaran akan lebih efektif (Kurniawan, 2017).

## 2.2.6 Jenis-Jenis Media Pembelajaran

Berikut ini adalah beberapa jenis media pembelajaran yang sering digunakan:

### a. Media Audio

Media audio adalah media yang isi pesannya hanya diterima melalui indera pendengaran. Dilihat dari sifat pesan yang diterima, media audio dapat menyampaikan pesan verbal (bahasa lisan atau kata-kata) maupun non verbal (bunyi-bunyian dan vokalisasi).

### b. Media Visual

Media visual adalah media yang mengandalkan indera pengelihatan. Media visual menampilkan materialnya dengan menggunakan alat proyeksi atau proyektor, karena melalui media ini perangkat lunak yang melengkapi alat proyeksi ini akan dihasilkan suatu cahaya atau gambar yang sesuai dengan materi yang diinginkan (Vikry, 2017).

## 2.2.7 Game

*Game* atau permainan merupakan aktivitas rekreasi dengan tujuan bersenang-senang mengisi waktu luang, atau berolahraga ringan. *Game* juga bisa diartikan sebagai arena keputusan dari aksi pemainnya karena ada target-target yang ingin dicapai, kelincahan intelektual (*intellectual agility*) pada tingkat tertentu juga merupakan ukuran sejauh mana *game* itu menarik untuk di mainkan secara maksimal (Kurniawan, 2017).

## 2.2.8 Sejarah Perkembangan Game

Asal usul permainan *video* atau *video game* terletak pada awal tabung sinar katoda berbasis pertahanan peluru kendali sistem pada akhir 1940-an. (Kurniawan, 2017). Pada tahun 1952, seorang mahasiswa Universitas *Cambridge* bernama A.S. Douglas membuat permainan OXO (*tic-tac-toe*) dalam versi grafik, permainan ini ia kembangkan ketika hendak mendemonstrasikan tesisnya tentang interaksi antar manusia dan computer. Pada tahun 1958 William Higin Botham mendesain sebuah *game* dengan judul *Tennis For Two* yang dimainkan dengan *oscilloscope*, dan kemudian ada pula Steve Russle pada tahun 1961 dengan *game* berjudul *Spacewar*



yang dibuat dalam computer *mainframe* DEC PDP-1 saat mereka menjalani studi di MIT.

Sejarah *video game* tidak hanya mengenai orang-orang yang berperan di dalamnya, tetapi juga mengenai berbagai perusahaan *game* yang mempunyai kasus ironis. *Atari* adalah perusahaan Amerika dengan Jepang, dan perusahaan Jepang bernama *SEGA* didirikan oleh orang Amerika. *Magnavox* yang memulai peran perusahaan ini telah berumur 1 abad, dan *Nintendo* sebagai perusahaan yang mempopulerkan kembali *video game* juga sama tuanya, serta tidak ada yang pernah berpikir bahwa *Sony* yang merupakan perusahaan penemu banyak barang-barang elektronik mulai dari transistor radio hingga alat perekam *video*, akan membuat sebuah *console* yang menjadi produk dengan tingkat penjualan yang tinggi hingga saat ini. Program-program ini kemudian diadaptasi ke dalam permainan sederhana lainnya di era dialami siswa, baik ketika berada di sekolah maupun dilingkungan rumah atau keluarganya sendiri (Dava, 2018).

### 2.2.9 Perkembangan Game

Berikut perkembangan game dari generasi ke generasi:

a. Game Generasi Pertama

Teknologi diciptakan pertama kali oleh A.S. Douglas pada tahun 1952 di *University of Cambridge*. Douglas mendemonstrasikan *game* buatannya melalui tesis dalam rangka mengenai interaksi antara komputer dan manusia. *Game* pertama yang ada di dunia ini ternyata sangat simpel, yakni sebuah permainan *Tic-Tac-Toe* atau XOXO yang diprogram menggunakan komputer EDSAC *vaccum-tube*, dimana komputer ini memiliki layar CRT (*Cathode Ray Tube*).

b. Game Generasi Kedua

Pada generasi kedua, teknologi *game* berkembang menjadi lebih baik dari sebelumnya. Masa ini dikenal dengan hadirnya grafik 8bit atau kurang lebih 4bit dalam sejarah komputer dan *video game*. Karena di tahun 1976, teknologi *game* berhasil dihidupkan kembali oleh Fairchild bersama karyanya yaitu VES (*Video Entertainment System*). Dalam dekade generasi kedua ini, banyak perusahaan yang menciptakan teknologi *game* berbasis konsol. Seperti

*Fairchild Channel F*, *Magnafax Odyssey* versi 2, *Attari 2600*, hingga *Attari 5200*. Pada era ini juga terlahir banyak *game-game* baru yang lebih mengenalkan suasana 3D seperti *Battlezone* buatan *Attari*, *Pac Man* keluaran *Namco*, *Game & Watch* milik *Nintendo* dan lain sebagainya dari perusahaan lainnya pula.

c. Game Generasi Ketiga

Konsol bernama *Famicon/Nintendo Entertainment System* (NES) yang dirilis di akhir tahun 1983 menjadi gebrakan baru yang diciptakan oleh perusahaan bernama *FAMICOM* di Jepang. Konsol yang diciptakan *FAMICOM* merupakan konsol yang pertama kali menampilkan gambar dan animasi resolusi tinggi. *FAMICOM* atau yang lebih dikenal dengan *Nintendo* ini memiliki chip pengaman pada *cartridge game* mereka, sehingga seluruh *game* yang akan dirilis harus dengan seizin *Nintendo*. Kemudian muncul *game* legendaris yang sampai saat ini masih memiliki banyak peminat, yaitu *Super Mario*.

d. Game Generasi Keempat

Teknologi *game* generasi empat disebut dengan generasi 16bit yang membawa perubahan pada *gameplay*, tata suara, grafik. Pada tahun 1988, *Nintendo* mendapatkan sambutan hangat oleh dunia. *SEGA* yang menyaingi *Nintendo*, merilis generasi konsol selanjutnya yaitu *Sega Mega Drive*. Kemudian *Nintendo* kembali mengeluarkan konsol generasi baru mereka pada tahun 1990, yaitu *SNES* (*Super Nintendo Entertainment System*).

Pada tahun yang sama *SEGA* mengeluarkan *Sega Megadrive* yang menyita perhatian para pencinta *game*, terutama Amerika. Kemudian *SEGA* melanjutkan produksi di tahun 1991 dengan *game Sonic the Hedgehog*. *Game* tersebut juga membuat para pencinta *game* karena kualitasnya yang jauh lebih baik daripada *Super Mario* ciptaan *Nintendo*.

e. Game Generasi Kelima

Pada tahun 1994, *Atari* kembali meluncurkan konsol untuk menyaingi konsol dari *SEGA* dan *Nintendo*. *Atari Jaguar* menyaingi kecanggihan *SNES* dan *Mega Drive*, tetapi penggunaannya yang sulit membuat tertarik pencinta *game*. *Game* generasi kelima ini *Sony* mengeluarkan konsol berbasis CD yang bernama

*Playstation*. Generasi ini juga bisa disebut dengan era konsol 32 bit. Konsol keluaran *Sony* menuai sukses untuk keluaran pertamanya yang kemudian menjadi konsol terlaris sepanjang masa.

f. Game Generasi Keenam

Teknologi game generasi enam masih dikuasai oleh *Sony*. Pada tahun 2000, *Sony* merilis *Playstation 2* yang berbasis DVD yang ukurannya lebih kecil. *Xbox* yang merupakan keluaran *Microsoft* adalah satu-satunya saingan *Playstation*. Meskipun tampilan *Xbox* sangat tajam dan berkualitas, tapi ternyata game *Xbox* tidak bisa mengalahkan populernya *Playstation 2*. Sampai saat ini *Sony* sudah meluncurkan *Playstation* versi 3, dan disusul versi ke 4.

g. Game Generasi Ketujuh

Pada generasi ini *SONY* mengeluarkan seri terbaru dari *PlayStation* yaitu *PlayStation 3* yang memiliki kualitas gambar lebih nyata. Disaat yang sama *Xbox 360* dan *Nintendo wii* keluar sebagai pesaing *PlayStation 3*. Semua konsol game ini memiliki kualitas gambar yang hampir setara namun sangat berbeda jauh dengan kualitas gambar pada generasi kelima. Data *PlayStation 3* misalnya, konsol ini menggunakan *Blue Ray* yang berkapasitas 200GB bahkan ada yang sampai 500GB. Semua konsol pada generasi ini dapat dimainkan pada PC atau komputer, namun dengan catatan komputer atau PC yang digunakan harus mempunyai spesifikasi yang mumpuni sehingga bisa menampilkan gambar yang bagus dan nyata (Sulistya, 2016).

## 2.2.10 Klasifikasi Game

Klasifikasi *game* dimaksudkan untuk memudahkan pengelompokan jenis *game*. Beberapa klasifikasi *game* adalah seperti berikut:

a. Berdasarkan Alat Yang Digunakan

- 1) ***Arcade Game*** atau juga yang sering disebut ding-dong di Indonesia, biasanya berada di daerah atau tempat khusus dan memiliki *box* atau mesin yang memang khusus di *design* untuk jenis *video games* tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih menikmati game tersebut, seperti pistol, kursi khusus, sensor gerakan,

sensor injakkan dan stir mobil beserta transmisinya. Untuk memainkan *game arcade* kita harus memasukkan uang logam terlebih dahulu.

- 2) **PC Game** adalah video *game* yang dimainkan menggunakan *Personal Computers*. Ada banyak sekali permainan yang bisa kamu mainkan. Sebelum bisa dimainkan, *game* ini harus kamu instal terlebih dahulu di PC kamu untuk bisa kamu nikmati. Biasanya, kontrol permainan bisa dari *keyboard* ataupun *mouse*.
- 3) **Konsol Game** merupakan alat *game* yang *portable*, biasa dibawa kemanapun. Contoh konsol *game* antara lain, *Sega*, *Playstation*, *Xbox*, *Nintendo Wii*. Yang dalam permainannya diperlukan *stick* untuk mengontrol *game* yang ingin dimainkan oleh pemain.
- 4) **Handheld Game** merupakan alat *game* yang biasa dibawa kemana-mana, ukurannya yang sesuai genggam tangan dengan sebuah monitor kecil di tengahnya. Contoh *handheld game* yaitu *Nintendo DS*, *Sony PSP*, *PS Vita*, *Steam deck*, dll.
- 5) **Mobile Game** adalah jenis *game* yang sekarang semakin marak berkembang untuk dimainkan di *smartphone*. *Mobile games*, yaitu *game* yang dapat dimainkan atau khusus untuk *mobile phone* dan tablet.

**b. Berdasarkan Jenis Permainannya**

- 1) **Adventure**, merupakan jenis permainan dimana dalam sebuah *game* tersebut kita diharuskan berjalan ke beberapa kota demi menjalankan sebuah misi. Dalam permainan jenis ini lebih mengutamakan sudut pandang ke 3 dibanding sudut pandang orang pertama dan memiliki tampilan 3D. *Game* jenis ini lebih mengutamakan alur cerita dan kemampuan berpikir para pemainnya dalam memecahkan teka-teki ataupun menyimpulkan peristiwa layaknya seorang detektif. Contoh *game* jenis ini adalah *Tomb Rider*, *Assassin Creed*, *Prince of Persia* dan masih banyak lagi.
- 2) **Casual Game**, Sesuai dengan namanya, *game* yang casual itu tidak kompleks, mainnya rileks dan sangat mudah untuk dipelajari (bahkan juga cenderung langsung bisa dimainkan). Jenis *game* ini biasanya memerlukan spesifikasi komputer yang standar pada jamannya dan ukurannya tidak

lebih dari 100 MB karena biasanya dapat di *download* versi demo-nya di *website* resminya. Genre permainannya biasanya *puzzle* atau *action* sederhana dan umumnya dapat dimainkan hanya menggunakan *mouse*.

- 3) **Education**, *game* bertipe edukasi yang dibuat dengan tujuan spesifik sebagai alat pendidikan, baik itu untuk belajar mengenal warna untuk balita, mengenal huruf dan angka, matematika, sampai belajar bahasa asing. *Developer* yang membuatnya harus memperhitungkan berbagai hal agar *game* ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya.
- 4) **Fighting**, *game* bertipe *fighting* pada dasarnya sama dengan *game* bertipe *action*, hanya saja *game* bertipe *fighting* pemain mengendalikan sebuah karakter untuk berkelahi dengan karakter lain sampai salah satu karakter kalah. Contoh *game* Janis ini adalah, *Tekken*, *Bloody Roar*, *Street Fighter*, *Mortal Kombat* dan sebagainya.
- 5) **MMO (Masive Multiplayer Online)**, merupakan *game* yang dapat dimainkan secara bersama-sama dengan menggunakan koneksi internet. Dalam *game* jenis ini mengutamakan kebersamaan dengan pemain lain. Contoh *game* jenis ini adalah, *Ragnarok Online*, *Rising Force (RF)*, *3 Kingdom*, *Gun Bound* dan sebagainya.
- 6) **Puzzle**, *game* jenis ini ditujukan untuk memecahkan suatu masalah tertentu. Hampir semua tantangan disini menyangkut masalah logika yang biasanya dibatasi oleh waktu. Contoh *game* jenis ini adalah, *The Room*, *Monument Valley*, *Tetris*, *Hitman Go* dan sebagainya.
- 7) **Racing**, merupakan jenis permainan balapan seperti balap mobil, motor, sepeda, *speed boat* atau yang lainnya. Dalam *game* jenis ini, kita diharuskan untuk dapat membalap lawan-lawan yang ada dihadapan kita demi menjadi juara. Contoh *game* jenis ini adalah, *Grand Turismo*, *Moto GP*, *Forza Horizon*, *Need For Speed* dan sebagainya.
- 8) **Real Time Strategy (RTS)**, *game* jenis ini melibatkan masalah strategi, taktik, dan logika. Contoh *game* jenis ini adalah, *Age of Empire*, *Strong Hold*, *Star Craft* dan sebagainya.

- 9) **Role Playing Game (RPG)**, merupakan jenis *game* yang melibatkan masalah taktik, logika, dan eksplorasi atau penjelajahan. Terkadang *game* jenis ini meliputi teka-teki dan masalah ekonomi karena pada *game* ini biasanya melibatkan pengumpulan barang-barang dan menjualnya untuk mendapatkan senjata yang lebih baik. Contoh dari *game* ini adalah, *Final Fantasy*, *Suikoden*, *Legend of Legaia* dan sebagainya.
- 10) **Shooting**, merupakan sebuah jenis permainan yang berlatarkan sebuah peperangan atau kehidupan mafia Dalam *game* jenis ini, dibutuhkan kecepatan refleks tangan yang cepat, mata yang tajam, dan timing yang pas agar dapat memenangkan sebuah pertarungan tembak-tembakan. Contoh dari *game* Janis ini adalah *Counter Strike*, *Call of Duty*, *Battlefield*, *Grand Theft Auto*, *The God Father* dan masih banyak lagi.
- 11) **Side Scrolling**, pada *game* jenis ini karakter dapat bergerak kesamping diikuti dengan gerakan *background*. Contoh *game* jenis ini adalah *Super Mario*, *Metal Slug*, *Megaman* dan sebagainya.
- 12) **Simulation**, merupakan jenis *game* yang menekankan pada kemampuan pemain dalam mengontrol permainan seperti, membangun sebuah kota, taman hiburan dengan pemain sebagai presiden atau pemilik taman hiburannya. Contoh *game* jenis ini adalah *The Sims City*, *Theme Park* dan sebagainya.
- 13) **Sport**, merupakan *game* yang menekankan olahraga dalam permainan, seperti voli, sepak bola, tenis, baseball dan sebagainya. Dalam *game* jenis ini pemain diharuskan memenangkan sebuah pertandingan dan menjadi juara. Contoh *game* jenis ini adalah *FIFA*, *Pro Evolutin Soccer*, *Winning Eleven* dan sebagainya.

#### 2.2.11 Game Edukasi

*Game* edukasi adalah permainan yang dirancang atau dibuat untuk merangsang daya pikir termasuk meningkatkan konsentrasi dan memecahkan masalah. *Game* edukasi adalah salah satu jenis media yang digunakan untuk memberikan pengajaran, menambah pengetahuan penggunanya melalui suatu media. Jenis *game* ini biasanya ditunjukan untuk anak-anak. Maka permainan

warna sangat diperlukan disini bukan tingkat kesulitan yang dipentingkan. Berdasarkan uraian diatas maka dapat disimpulkan game edukasi adalah salah satu bentuk game yang dapat berguna untuk menunjang proses belajar-mengajar secara lebih menyenangkan dan lebih kreatif, dan digunakan untuk memberikan pengajaran atau menambah pengetahuan penggunanya melalui suatu media yang menarik (Handriyantini, 2009).

#### **2.2.12 Matematika**

Matematika Secara bahasa (lughowi), kata “Matematika” berasal dari bahasa Yunani yaitu “Mathema” atau mungkin juga “Mathematikos” yang artinya hal-hal yang dipelajari. Matematika suatu alat untuk mengembangkan cara berfikir. Menurut Andi Hakim Nasution (dalam Masykur, 2008) “penggunaan istilah matematika lebih tepat daripada ilmu pasti. Karena dalam matematika, banyak terdapat pokok bahasan yang justru tidak pasti seperti pada statistik terdapat probabilitas (kemungkinan)”.

Matematika secara umum didefinisikan sebagai bidang ilmu yang mempelajari pola dari struktur, perubahan dan ruang. Maka secara informal dapat juga di sebut sebagai ilmu bilangan dan angka. Dalam pandangan formalis, matematika adalah penelaahan struktur abstrak yang didefinisikan secara aksioma dengan menggunakan logika simbolik dan notasi. Adapun pandangan lain bahwa matematika adalah ilmu dasar yang mendasari ilmu pengetahuan lain (Ramadhani & Siregar, 2022)

#### **2.2.13 Android**

Android adalah suatu sistem operasi yang didesain sebagai platform *open source* untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android menyediakan semua tools dan framework untuk mengembangkan aplikasi dengan mudah dan cepat. Dengan adanya Android SDK (*Software Development Kit*) pengembang aplikasi dapat memulai pembuatan aplikasi pada platform Android menggunakan bahasa pemrograman Java (Busran & Fitriyah, 2015).

Sistem operasi Android ini bersifat *open source* sehingga programmer berbondong-bondong untuk membuat aplikasi maupun memodifikasi sistem ini. Para programmer memiliki peluang yang sangat besar untuk terlibat mengembangkan aplikasi Android karena alasan *open source* tersebut. Sebagian besar aplikasi yang terdapat dalam *Play Store* bersifat gratis dan ada juga yang berbayar.

#### **2.2.14 Kelahiran Android**

Perkembangan *Android* dimulai dengan berdirinya *Android Inc.* pada Oktober 2003 dengan tujuan *mobile device* yang lebih pintar untuk menyaingi *Symbian* dan *Windows Mobile* yang populer pada saat itu dimana *iPhone* dan *Blackberry* belum dirilis. *Android Inc* yang merupakan perusahaan startup milik Andy Rubin berhasil ditemukan oleh raksasa pencari Google yang kemudian sepakat untuk memberikan dukungan finansial dan membelinya pada Agustus 2005 hingga akhirnya dijadikan anak perusahaan yang sepenuhnya dimiliki oleh Google.

Bersamaan dengan itu, terbentuklah sebuah konsorsium bernama *Open Handset Alliance* yang terdiri atas 34 perusahaan yang bergerak di bidang perangkat keras, perangkat lunak, serta telekomunikasi, anggotanya meliputi: Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia yang saling bekerja sama untuk memajukan standar sumber terbuka (*Open Source*) termasuk sistem operasi Android itu sendiri pada perangkat seluler. Pada tahun 2005, *Android* diakuisisi oleh Google, pengembangan terus dilanjutkan sampai *Android* versi *beta* diluncurkan pada tanggal 5 November 2007. Hingga saat ini tanggal 5 November diperingati sebagai hari jadi *Android*. Seminggu setelahnya yaitu pada tanggal 12 November 2007 *Android SDK (Software Development Kit)* diluncurkan, sehingga pengguna dapat membuat dan mengembangkan aplikasi *Android* mereka sendiri (Satyaputra & Aritonang, 2014)

#### **2.2.15 Perkembangan Android**

Hingga pada akhirnya smartphone pertama kali yang menggunakan sistem operasi Android pun dirilis, yakni tepatnya pada tanggal 22 Oktober 2008 telah diluncurkan smartphone bersistem operasi Android buatan HTC Corporation yang



bernama HTC Dream. Semenjak muncul terobosan tersebut, banyak vendor lain yang ikut merilis ponsel dengan platform Android di produk-produk ponsel mereka. Hingga sekarang pun makin banyak vendor-vendor bermunculan untuk saling gencar membuat dan merilis berbagai macam smartphone guna meraih peluang besar dalam pasar-pasar smartphone Android. (Madcoms, 2018).

Sistem operasi *Android* ini sangatlah unik dan mampu memberikan kemudahan bagi para pengguna karena nama sistem operasinya selalu berdasarkan nama makanan dan diawali dengan abjad yang berurutan seperti pada tabel berikut ini.

Berikut adalah daftar versi dari *Android*:

a. Android Beta

Pertama kali dirilis pada 5 November 2007, kemudian pada 12 November 2007 menggunakan aplikasi bernama Software Development Kit (SDK) yang dirilis oleh Google. (Madcoms, 2018).

b. Android 1.0

Pertama kali dirilis pada 23 September 2008. Sebenarnya Android versi perdana ini awalnya dinamai ASTRO tapi karena alasan hak cipta atas nama ASTRO itu sendiri, nama ASTRO batal disematkan pada versi pertama dari sistem operasi Android ini lalu diubah ke nama ALPHA. HTC Dream adalah ponsel pertama yang menggunakan Android versi ini (Madcoms, 2018).

c. Android 1.1

Pertama kali dirilis pada 9 Februari 2009. Versi Android kedua ini juga mengalami masalah hak cipta penamaan yang sama dengan versi pertamanya. Pada awalnya, Android ini akan diberi nama BENDER akan tetapi karena alasan melanggar hak cipta, nama BENDER batal disematkan pada versi Android ini lalu diubah ke nama BETA. Versi Android ini dirilis pertama kali untuk perangkat ponsel bernama T-Mobile G1. Versi ini merupakan pembaruan untuk memperbaiki beberapa bug atau kesalahan sistem, mengganti API sekaligus menambahkan beberapa fitur dari versi Android sebelumnya (Madcoms, 2018).

d. Android 1.5 "*Cupcake*"

Pertama kali dirilis pada 30 April 2009. Mulai versi Android ini penamaannya menggunakan nama makanan manis sebagai pencuci mulut (dessert) mulai

digunakan karena ini merupakan versi rilis yang ketiga maka penamaanya diawali dengan huruf “C” sesuai dengan urutan abjad dan jadilah nama “*Cupcake*” menjadi nama resmi dari Android ketiga ini. Android ini berbasis pada kernel Linux 2.6.27 dan menambahkan beberapa update serta *User Interface* (UI) baru dari versi Android sebelumnya, Mulai terdapat komponen *widget* yang dapat diatur ukurannya, kemudian ditambah kemampuan untuk mengunggah video dan gambar ke Youtube serta layanan berbagi foto Picasa (Madcoms, 2018).

e. Android 1.6 "*Donut*"

Dirilis pertama kali pada 15 September 2009. Terdapat peningkatan pada fitur pencarian dan UI yang lebih menarik. Pada versi ini juga sudah mendukung teknologi jaringan CDMA/EVDO, 802.1x, VPNs. Selanjutnya dukungan layar dengan resolusi WVGA (Madcoms, 2018).

f. Android 2.0/2.1 "*Eclair*"

Diliris pertama kali 9 Desember 2009. Terdapat penambahan fitur untuk mengoptimalkan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan lampu flash untuk kamera 3.2 MP, digital zoom, dan Bluetooth 2.1. beberapa versi updatenya antara Android v.2.0 kemudian v2.0.2 dan terakhir v.2.1 (Madcoms, 2018).

g. Android 2.2 "*Froyo*"

Dirilis pertama kali pada 20 Mei 2010 pada smartphone Google Nexus One. Pada versi ini sudah mendukung terhadap Adobe Flash Player 10.1. Peningkatan pada kecepatan membuka serta menutup aplikasi. Serta penggunaan SD Card sebagai tempat penyimpanan aplikasi. Ketika Android Froyo hadir, mulai muncul banyak diskusi yang membahas mengenai persaingan antara Android dengan iOS. Beberapa versi update yang dirilis antara lain Android v.2.2.1 hingga v2.2.3 (Madcoms, 2018).

h. Android 2.3 "*Gingerbread*"

Pertama kali diperkenalkan pada 6 Desember 2010. Terjadi banyak peningkatan pada versi Android yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalkan kemampuan aplikasi serta permainan. Serta

mulai digunakan Near Field Communication (NFC). Perbaikan terhadap dukungan layar resolusi WXGA dan di atasnya. Beberapa versi update yang dirilis antara lain v2.3.3 hingga v2.3.7 (Madcoms, 2018).

i. Android 3.0/3.1 "*Honeycomb*"

Pertama kali diperkenalkan pada 22 Februari 2011 dan Motorola Xoom adalah yang pertama kali menggunakannya. Android versi ini merupakan versi yang didesain khusus untuk pengoptimalan penggunaan pada perangkat tablet (Madcoms, 2018).

j. Android 4.0 "*Ice Cream Sandwich*"

Pertama kali dirilis pada 19 Oktober 2011. Smartphone yang pertama kali menggunakan versi Android ini adalah Samsung Galaxy Nexus. Secara teori semua perangkat seluler yang menggunakan versi Android sebelumnya yaitu Android Gingerbread dapat diperbarui ke Android Ice Cream Sandwich (Madcoms, 2018).

k. Android 4.1/4.2/4.3 "*Jelly Bean*"

Google mengumumkan Android 4.1 dalam konferensi Google I/O pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja UI. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7 yang dirilis pada 13 Juli 2012 (Madcoms, 2018).

l. Android 4.4 "*Kitkat*"

Android Kitkat rilis pada tanggal 3 September 2013. Meskipun pada awalnya diberi nama "Key Lime Pie" nama itu berubah karena sangat sedikit orang benar-benar tahu rasa makanan manis tersebut. Beberapa blogger teknologi juga mengharapkan rilis "Key Lime Pie" menjadi Android 5. Kitkat memulai debutnya pada Google Nexus 5 pada tanggal 31 Oktober 2013, dan dioptimalkan untuk berjalan pada rentang yang lebih besar dari perangkat versi Android sebelumnya, memiliki 512 MB RAM sebagai standar minimum yang disarankan (Madcoms, 2018).

m. Android 5.0/5.1 "*Lollipop*"

Android Lollipop adalah versi stabil terbaru dengan versi antara 5.0 dan 5.1. diresmikan pada 25 Juni 2014 saat konferensi Google I/O dan tersedia secara

resmi melalui pembaruan online pada tanggal 12 November 2014, untuk memilih perangkat yang menjalankan distribusi Android dilayani oleh Google (seperti perangkat Nexus dan Google Play edition). Salah satu perubahan yang paling menonjol dalam rilis. Lollipop adalah UI yang didesain ulang dan dibangun dengan gaya material design. Perubahan lain termasuk perbaikan pemberitahuan, yang dapat diakses dari layar kunci dan ditampilkan pada banner di bagian atas layar. Google juga membuat perubahan internal untuk platform, dengan Android Runtime (ART) secara resmi menggantikan Dalvik untuk meningkatkan kinerja aplikasi serta dengan perubahan yang ditujukan untuk meningkatkan dan mengoptimalkan penggunaan baterai smartphone (Madcoms, 2018).

n. Android 6.0 "*Marshmallow*"

Android Marshmallow merupakan versi Android yang terbilang baru saat buku ini ditulis. Android ini rilis perdana pada tanggal 5 Oktober 2015, versi Android ini membawa segudang fitur baru diantaranya dukungan native pengenalan tanda sidik jari, sistem manajemen daya baterai bernama Doze, serta dukungan terhadap konektor USB tipe C. Pada Juli 2017 sebanyak 31,8% dari jumlah seluruh perangkat di dunia bersistem Android Marshmallow (Madcoms, 2018).

o. Android 7.0 "*Nougat*"

Nougat adalah versi Android termutakhir yang baru diperkenalkan pada ajang konferensi Google I/O, pertengahan tahun 2016. Beberapa lama setelahnya, Google menghadirkan Nougat secara resmi untuk publik. Pembaruan paling mendasar pada versi Nougat adalah kehadiran Google Assistant yang menggantikan Google Now. Asisten digital tersebut lebih bisa diandalkan untuk menjalankan berbagai macam fungsi. Fitur-fitur baru lainnya mencakup layar split-screen saat dipakai multitasking, serta fitur Doze yang telah dikenalkan di versi Android Marshmallow sebelumnya namun telah ditingkatkan. Android Nougat juga memiliki dukungan terhadap platform virtual reality terbaru dari Google (Madcoms, 2018).

p. Android 8.0 "*Oreo*"

Android Oreo merupakan versi ke-8 dari pengembangan sistem operasi Android yang telah dirilis ke publik secara resmi pada tanggal 21 Agustus 2017.

Perangkat yang mendukung sistem operasi Android Oreo adalah Google Pixel (Madcoms, 2018).

q. Android 9.0 "*Pie*"

Android Pie adalah pembaruan selanjutnya dari Android 8.1 "Oreo" yang merupakan versi pembaruan urutan ke-16 dari sistem operasi Android, untuk versi pengembangan yang pertama kali dirilis pada Maret 2018 dan dirilis ke publik pada 6 Agustus 2018 (Madcoms, 2018).

r. Android 10

Android 10 adalah versi kesepuluh dari sistem operasi Android. Pertama kali diumumkan oleh Google pada 13 Maret 2019, dan beta pertama dirilis pada hari yang sama dengan nama "Android Q". Beta kedua dirilis pada 3 April 2019. Beta ketiga dirilis pada konferensi Google I/O pada 7 Mei 2019. Kemudian, pada 5 Juni, Google merilis beta keempat, dengan API final dan SDK (API Level) 29). Pada 10 Juli 2019, Google merilis Beta 5 dengan API 29 SDK terakhir serta optimisasi terbaru dan perbaikan bug. Pada 7 Agustus 2019, Google merilis Beta 6, yang merupakan beta terakhir sebelum pengumuman Android Q. Pada 22 Agustus 2019, Google mengumumkan bahwa versi rilis resmi akan disebut hanya Android 10.

s. Android 11

Siklus penamaan menggunakan nama dessert berhenti setelah Google merilis Android 10. Dirilis pada 2020, ada sejumlah fitur Android 11 yang bisa kamu nikmati setelah versi ini dirilis, misalnya fitur keamanan yang lebih canggih dari sebelumnya. Selain itu, ada banyak fitur menarik lainnya dari Android 11. Seperti notifikasi, *screen recorder* tanpa aplikasi tambahan, picture-in-picture, hingga mengambil *screenshot* panjang.

t. Android 12

Google baru saja meluncurkan Android 12 pada tanggal 18 Mei 2021 lalu. Sistem operasi ini masih dalam tahap pengembangan dan dinamakan sebagai Android 12 beta preview. Android 12 menawarkan beragam fitur menarik yang belum ada pada pendahulunya, seperti perubahan desain *Material You* yang memudahkan kostumasi handphone, peningkatan fitur privasi, dan mode satu tangan. Peningkatan fitur privasi dilakukan dengan adanya *Privacy Dashboard*

sebagai panel utama kita saat memantau informasi penggunaan data pribadi seperti akses lokasi, kamera, mikrofon dalam 24 jam terakhir dari aplikasi apapun.

#### 2.2.16 Android SDK (Software Development Kit)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*.

Beberapa fitur Android yang penting adalah sebagai berikut:

- a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- b. DVM dioptimalkan untuk perangkat *mobile*.
- c. *Integrated browser* berdasarkan *engine open source WebKit*.
- d. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *OpenGL ES 1.0*.
- e. SQLite untuk penyimpanan data.
- f. Dukungan untuk audio, video dan gambar.
- g. *Bluetooth, Edge, 3G, Wifi*.
- h. Kamera, *GPS*, kompas dan *accelerometer*.

Lingkungan development yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging, profil* dan kinerja memori serta *plugins* untuk IDE Android Studio (Kusniyati & Sitanggang, 2016).

#### 2.2.17 UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah sebuah bahasa pemrograman yang berdasarkan grafik atau gambar untuk memvisualisasi, memspesifikasi, membangun, dan mendokumentasi dari sebuah sistem pengembangan *software* berbasis OO (*Object Oriented*).

UML merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industry pengembang sistem atau perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan raksasa seperti IBM, Microsoft, dan sebagainya (Hamim, 2014).

**Table 2. 1 Jenis-Jenis Diagram UML**

No	Diagram	Kegunaan
1	<i>Activity</i>	Prosedur perilaku dan paralel.
2	<i>Class</i>	Kelas fitur dan hubungan-hubungan.
3	<i>Communication</i>	Interaksi antara objek, penekanan pada jalur.
4	<i>Component</i>	Struktur dan koneksi komponen.
5	<i>Composite structure</i>	Dekomposisi waktu jalan sebuah kelas.
6	<i>Deployment</i>	Pemindahan artifak ke node.
7	<i>Interaction</i>	Campuran sequence dan activity diagram.
8	<i>Object</i>	Contoh konfigurasi dari contoh-contoh
9	<i>Package</i>	Struktur hirarki dari compile – time.
10	<i>Sequence</i>	Interaksi antar objek, penekanan pada <i>sequence</i> .
11	<i>State machine</i>	Bagaimana <i>even</i> mengubah objek selama aktif.
12	<i>Timing</i>	Interaksi antar objek, penekanan pada <i>timing</i> .
13	<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sebuah sistem

Dalam penggunaan UML pada sebuah sistem terdapat beberapa tujuan, sebagai berikut:

- a. Memberikan model yang siap pakai, bahasa permodelan yang visual yang ekspresif

- b. Menciptakan suatu bahasa permodelan yang menggunakan konsep berorientasi objek.
- c. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.

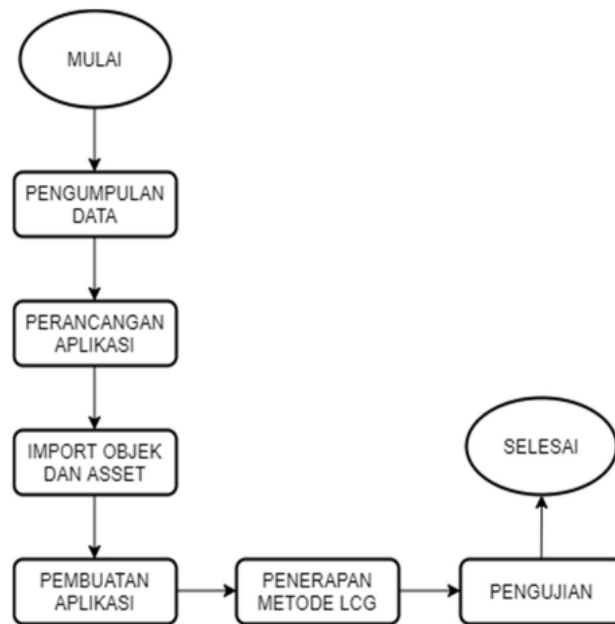
Dalam pembuatan UML terdapat beberapa alat bantu dalam bentuk diagram Seperti, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*. Diagram-diagram ini merupakan *artefact* utama UML. *Data-Flow-Diagram* dan tipe diagram lain yang tidak terdapat dalam UML tidak termasuk dalam paradigm konsep berorientasi objek. *Activity diagram* dan *collaboration diagram* yang terdapat dalam UML menggantikan *data-flow-diagram*. *Activity diagram* juga bermanfaat untuk membuat *work-flow* (Hamim, 2014).

### 2.3 Kerangka Penelitian

Untuk merancang aplikasi yang ingin dibuat, perancang membuat tahapan-tahapan yang akan dilakukan untuk menyelesaikan aplikasi ini, diantaranya:

1. Tahap Pengumpulan Data, tahapan ini dimulai dengan mencari informasi mengenai materi yang akan dibuat dan mencari kebutuhan perangkat atau tools apa saja yang akan digunakan.
2. Tahap Perancangan, tahapan ini digunakan untuk merancang aplikasi agar sesuai dengan konsep yang sudah ditentukan.
3. Tahap Import Objek dan Asset, setelah selesai didesain maka selanjutnya objek dan asset tersebut diimport kedalam tools yang digunakan.
4. Tahap Pembuatan Aplikasi, tahap ini dimulai dengan menggabungkan rancangan- rancangan dan objek-objek yang telah dibuat sebelumnya.
5. Tahap Penerapan Metode LCG, metode ini akan diterapkan kedalam aplikasi yang telah selesai dibuat.
6. Tahap Pengujian, setelah aplikasi selesai dibuat maka akan dilakukan pengujian untuk mengetahui adanya masalah yang muncul atau tidak.



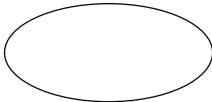
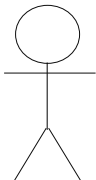




**Gambar 2. 1 Flowchart Kerangka Penelitian**

## 2.4 Use Case Diagram

*Use Case* diagram digunakan untuk menjelaskan apa yang akan dilakukan oleh sistem serta aktor yang berhubungan dengan proses yang ada pada sistem. Pada perancangan *use case* diagram ini akan menjelaskan hal-hal yang dapat dilakukan *user* atau pengguna Aplikasi *Game* Edukasi Berbasis *Mobile*.

**Table 2. 2 Simbol-Simbol *Use Case* Diagram**


Simbol	Keterangan
	<b><i>Use Case</i></b> Menggambarkan fungsionalitas system.
	<b><i>Actor</i></b> Menggambarkan sesuatu (entitas) yang berhubungan dengan system dan berpartisipasi dalam <i>use case</i> .


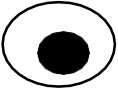

	<p><b>Asosiations</b></p> <p>Menggambarkan bagaimana hubungan antara <i>use case</i> dengan <i>use case</i>, <i>actor</i> dan <i>actor</i> atau antara <i>actor</i> dan <i>use case</i>.</p>
	<p><b>Package</b></p> <p>Digunakan ketika diagram mempunyai banyak <i>bubbles</i> dan tidak bias dicetak dalam satu halaman.</p>

## 2.5 Activity Diagram

*Activity Diagram* mempunyai manfaat dalam membantu memahami suatu proses secara keseluruhan apabila *activity diagram* ini dibuat terlebih dahulu sebelum memodelkan suatu proses. Dalam artian yang lain *activity diagram* dapat berupa gambaran suatu proses, dipakai pada bisnis modeling untuk memperlihatkan urutan aktifitas proses bisnis, memiliki struktur diagram yang mirip *flowchart* atau data flow diagram pada perancangan terstruktur. *Activity diagram* ini bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan *activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*. Dan berikut adalah tampilan dari *activity diagram* yang ada pada Aplikasi Game Edukasi.

**Table 2. 3 Notasi Activity Diagram**

Notasi	Keterangan
	<p><b>State</b>, digambarkan berbentuk persegi empat dengan sudut membulat dan memiliki nama sesuai dengan kondisinya saat itu.</p>

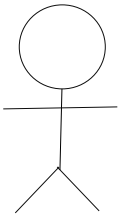
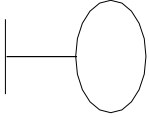
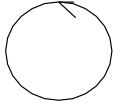
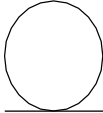



	<b>Awal (<i>Start</i>)</b> , digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram <i>flowchart</i> .
	<b>Titik Akhir (<i>End</i>)</b> , digunakan untuk menggambarkan akhir dari kejadian dalam suatu diagram <i>flowchart</i> .
[guard]	<b><i>Guard</i></b> , yang merupakan syarat terjadinya transisi yang bersangkutan.
	<b><i>Point</i></b> , digunakan untuk menggambarkan apakah akan masuk ( <i>point enter</i> ) ke dalam <i>state</i> atau akan keluar ( <i>exit point</i> )
<i>Event</i>	<b><i>Event</i></b> , digunakan untuk mendeskripsikan kondisi yang menyebabkan sesuatu pada <i>state</i> .

## 2.6 Sequence Diagram

*Sequence* Diagram menggambarkan objek yang berpartisipasi dalam sebuah *use case* dan pesan yang terdapat di antara mereka dari waktu ke waktu untuk suatu *use case*, *sequence* diagram juga dipakai untuk memodelkan deskripsi tentang sistem yang ada pada sebuah atau beberapa *use case* pada *use case* diagram, yang menggambarkan hubungan antara aktor dan *use case*. Berikut ini adalah *sequence* diagram yang terdapat pada aplikasi.

**Table 2. 4 Simbol *Sequence Diagram***

Simbol	Keterangan
--------	------------

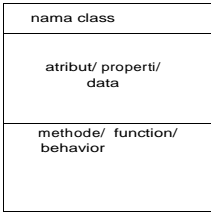

	<p><b><i>Actor Object</i></b></p> <p>Menggambarkan pihak yang melakukan interaksi atau yang memicu sistem untuk berfungsi.</p>
	<p><b><i>Boundary Object</i></b></p> <p>Menggambarkan obyek yang menjadi kontrol sistem.</p>
	<p><b><i>Control Object</i></b></p> <p>Menggambarkan obyek yang menjadi kontrol sistem.</p>
	<p><b><i>Entity Object</i></b></p> <p>Menggambarkan obyek yang berupa entitas.</p>
	<p><b><i>Lifeline</i></b></p> <p>Menggambarkan eksekusi obyek selama <i>sequence</i></p>
	<p><b><i>Message</i></b></p> <p>Menggambarkan komunikasi yang terjadi antar obyek</p>
	<p><b><i>Activation</i></b></p> <p>Menunjukkan periode selama suatu obyek atau actor sedang melakukan suatu tindakan.</p>

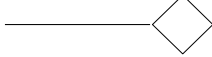
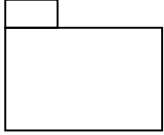
## 2.7 Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain (Reni, 2017).

*Class* dibagi menjadi tiga bagian, nama kelas, atribut kelas, serta operasi kelas. Nama kelas adalah nama dari kelas itu sendiri, misalnya kelas pelajar, kelas guru, kelas wali murid, dan lain-lain. Penamaan kelas menggunakan kata benda. Atribut adalah data yang dimiliki oleh kelas tersebut, misalnya kelas siswa memiliki atribut nomor induk, nama, tanggal lahir, alamat, jenis kelamin, dan lain-lain. Lalu operasi kelas menunjukan apa yang kelas tersebut bisa lakukan, misalnya kelas guru dapat melakukan operasi mengajar, absensi, penelitian, dan lain sebagainya (Fitri, 2019).

**Table 2. 5 Simbol *Class Diagram***

Simbol	Keterangan
	<b><i>Class</i></b> Menggambarkan himpunan object sejenis yang mempunyai atribut dan metode yang sama.
	<b><i>Asociation</i></b> Menggambarkan hubungan yang terjadi antar <i>class</i> .

	<p><b>Agregasi</b></p> <p>Menggambarkan suatu <i>class</i> terdiri dari <i>class</i> lain atau suatu kelas adalah bagian dari kelas lain.</p>
	<p><b>Package</b></p> <p>Digunakan untuk mengelompokkan <i>class</i> yang mempunyai persamaan.</p>

## 2.8 RAD (Rapid Application Development)

*Rapid Application Development* RAD adalah model proses perkembangan *software* yang menekankan pada daur pengembangan hidup yang singkat. Model RAD ini merupakan sebuah adaptasi cepat dari model *waterfall* dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi komponen. RAD merupakan gabungan dari bermacam-macam teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application* untuk mempercepat pengembangan sistem/aplikasi. Dari definisi konsep RAD ini, dapat dilihat bahwa pengembangan aplikasi dengan menggunakan metode RAD dapat dilakukan dalam waktu yang relatif lebih cepat. Sesuai dengan metodologi RAD (Hariyanto, Sastra, & Putri, 2021).

RAD digunakan pada aplikasi sistem konstruksi, maka menekankan tahap-tahap. berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi.



**Gambar 2. 2 *Rapid Application Development***

Tahapan RAD terdiri dari 3 tahap yang terstruktur dan saling bergantung disetiap tahap, yaitu:

1. **Requirement Planning**, dalam tahap ini diketahui apa saja yang menjadi kebutuhan system yaitu dengan mengidentifikasi kebutuhan informasi dan masalah yang dihadapi untuk menentukan tujuan, batasan-batasan system, kendala dan juga alternatif pemecahan masalah. Analisis digunakan untuk mengetahui perilaku system dan juga untuk mengetahui aktivitas apa saja yang ada dalam system tersebut.
2. **Design Workshop**, yaitu mengidentifikasi solusi alternatif dan memilih solusi yang terbaik. Kemudian membuat desain proses bisnis dan desain pemrograman untuk data-data yang telah didapatkan dan dimodelkan dalam arsitektur system informasi. *Tools* yang digunakan dalam pemodelan system biasanya menggunakan *Unified Modeling Language (UML)*.
3. **Implementation**, setelah *Design Workshop* dilakukan, selanjutnya system diimplementasikan (*coding*) ke dalam bentuk yang dimengerti oleh mesin yang diwujudkan dalam bentuk program atau unit program. Tahap implementasi system merupakan tahap meletakkan system supaya siap untuk dioperasikan.

## 2.9 Algoritma

Algoritma adalah logika, suatu metode khusus yang tepat dan terdiri dari serangkaian tahapan/urutan yang terstruktur dan dituliskan secara matematis yang digunakan untuk memecahkan suatu permasalahan. Algoritma dapat juga diartikan sebagai urutan system secara sistematis dan logis. Dalam perkembangannya, algoritma merupakan bentuk dasar dari perintah-perintah yang akan di-Coding-kan kedalam form yang telah dirancang pada tahap implementasi system yang banyak dipakai di bidang computer (Manurung, 2013).

## 2.10 Metode Linear Congruential Generator

Metode *Linear Congruential Generator* (LCG) digunakan untuk menghasilkan bilangan acak yang digunakan dalam permainan *Solve The Math* ini. Bilangan acak yang dihasilkan oleh Linear Congruential Generator (LCG) ini akan digunakan untuk menentukan posisi dari bilangan 1 sampai n pada setiap baris.

Metode *Linear Congruential Generator* (LCG) merupakan salah satu algoritma pengacakan yang digunakan untuk menghasilkan bilangan acak (*Random Number*). Model matematis LCG dapat dihitung dengan menggunakan persamaan:

$$x_{n+1} = ((a \times x_n) + b) \bmod m$$

Keterangan :

$x_{n+1}$  = bilangan acak ke-n dari deretnya

$x_n$  = bilangan acak sebelumnya

$a$  = faktor pengali

$b$  = penambah

$m$  = jumlah soal

$n$  = 0, 1, 2, 3, . . . dan seterusnya

Pengacakan soal dengan metode LCG dilakukan melalui empat tahapan. Penentuan jumlah soal yang diacak ( $m$ ) merupakan tahap pertama yang dilakukan dalam penelitian ini. Setelah itu dilakukan tahap penentuan nilai variabel faktor pengali ( $a$ ). Tahap ketiga merupakan tahap penentuan variabel penambah ( $b$ ). Tahap akhir pengacakan soal dengan metode LCG adalah perhitungan kombinasi dari variabel, dan dengan menggunakan persamaan.

## 2.11 Aplikasi Pendukung

Aplikasi pendukung adalah beberapa software yang digunakan untuk mendukung proses pembuatan aplikasi yang akan dibangun oleh penulis.

## 2.12 Android Studio

Android studio adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi Android, pemrograman Android resmi dari Google yang dibangun diatas perangkat lunak *JetBrains IntelliJ IDEA* dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android.



Android Studio menampilkan file dalam tampilan proyek Android. Tampilan ini disusun oleh modul untuk menyediakan akses cepat ke file sumber utama proyek. Semua file build terlihat di tingkat atas di bawah *Gradle Scripts* dan setiap modul aplikasi berisi folder Bermanifestasi, Java, dan Res.



**Gambar 2. 3 Android Studio**

### **2.13 Unity 3D**

Unity 3D adalah sebuah game engine yang berbasis cross-platform. Unity dapat digunakan untuk membuat sebuah game yang bisa digunakan pada PC, Android, iPhone, PS3, dan bahkan X-BOX. Unity adalah sebuah tool yang terintegrasi untuk membuat game, arsitektur bangunan dan simulasi. Unity bisa untuk games PC dan games Online. Untuk games Online diperlukan sebuah plugin, yaitu Unity Web Player, sama halnya dengan Flash Player pada Browser. Unity tidak dirancang untuk proses desain atau modelling, dikarenakan unity bukan tool untuk mendesain. Jika ingin mendesain, pergunakan 3D editor lain seperti 3dsmax atau Blender. Banyak hal yang bisa dilakukan dengan unity, ada fitur *audio reverb zone*, *particle effect*, dan *Sky Box* untuk menambahkan langit. Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo. *Flexible* dan *EasyMoving*, *rotating*, dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating*, *removing*, dan *changing properties*. *Visual Properties Variables* yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di drag and drop, bisa memilih warna dengan color picker berbasis .NET. Artinya penjalanan program dilakukan dengan OpenSource .NET platform, Mono.



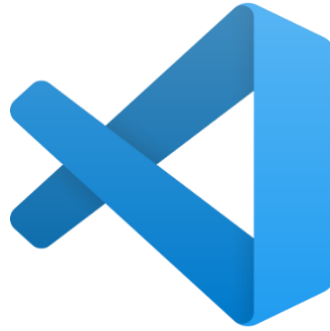
**Gambar 2. 4 Unity 3D**

## **2.14 Visual Studio Code**

*Visual studio code* merupakan sebuah teks editor yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang melalui *marketplace Visual Studio Code* (seperti C++, C#, Python, Go, Java, dan masih banyak lagi).

Seperti editor pada umumnya *VSCode* memiliki fitur *syntax coloring* dan *bracket matching*. *VSCode* juga memiliki fitur *IntelliSense* yang cukup canggih. Fitur ini akan sangat terasa apabila kita mengembangkan aplikasi web yang menggunakan HTML, CSS, JavaScript, JSON, Less, atau SASS. Fitur *IntelliSense* merupakan salah satu fitur yang membantu kita dalam menulis kode program melalui *popup* yang muncul secara otomatis pada saat kita sedang mengetik dan menampilkan saran sintaks yang bisa kita pilih.

Teks editor *VSCode* bersifat *open source*, yang mana kode sumbernya dapat pengguna lihat dan pengguna dapat berkontribusi untuk pengembangannya. Hal ini juga yang membuat *VSCode* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *VSCode* ke depannya.



**Gambar 2. 5 Visual Studio Code**

## **2.15 Bahasa Pemrograman**

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi (Richy, 2018).

Bahasa Pemrograman (programming language) adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi (Richy, 2018).

Bahasa pemrograman merupakan komponen yang penting dalam membuat sebuah program atau aplikasi. Berikut beberapa contoh dari bahasa pemrograman

### **2.15.1 C#**

C# atau yang dibaca C sharp merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka *.NET Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek - aspek atau pun fitur bahasa yang terdapat

pada bahasa - bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic* dan lain - lain dengan beberapa penyederhanaan. Menurut standar ECMA-334 *C# Language Specification*, nama *C#* terdiri atas sebuah huruf latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat didalam keyboard standar (Jonathan, 1998).

Dalam mengembangkan aplikasi ataupun fungsi-fungsi lain yang berbasis *C#*, kita perlu IDE (*Integrated Development Environment*) untuk membantu pekerjaan kita. Di Windows kita dapat menggunakan Visual Studio yang memiliki fitur yang sangat lengkap. Dalam pemograman *C#* memiliki 5 struktur dasar, yaitu:

a. Resourch atau library

Struktur pertama ini merupakan pendefinisian library apa yang harus ada pada program kita atu library apa yang kita impor.

b. Namespace

Struktur kedua ini adalah nama dari project kita.

c. Nama Class

Struktur ketiga ini tentang apa nama dari Class yang kita buat dan bisa juga langsung diberi penanda seperti Main Class yang menandakan bahwa Class tersebut Class utama.

d. Deklarasi Method

Struktur keempat ini merupakan pendeklarasian method sebagai awalan untuk menjalankan method atau perintah yang ada di dalamnya, jika didefinisikan dengan “Main” maka method tersebut yang dijalankan pertama kali oleh computer.

e. Method atau Command

struktur kelima adalah method atau perintah yang kita berikan untuk di eksekusi oleh compiler.



**Gambar 2. 6 Bahasa Pemrograman C#**

## **2.16 Metode Pengujian**

Pengujian merupakan hal terpenting yang bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang akan diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

### **2.16.1 Tujuan Pengujian**

Tujuan dalam pengujian dari sebuah perangkat lunak adalah sebagai berikut:

- a. Proses menjalankan program dengan maksud untuk mencari kesalahan (*error*).
- b. Kasus uji yang baik adalah kasus yang memiliki peluang untuk mendapatkan kesalahan yang belum diketahui.
- c. Pengujian dikatakan berhasil bila dapat memunculkan kesalahan yang belum diketahui.

Jadi pengujian yang baik bukan untuk memastikan tidak ada kesalahan, tetapi untuk mencari sebanyak mungkin kesalahan yang ada pada program.

### **2.16.2 Prinsip Pengujian**

Perinsip dalam pengujian dari sebuah perangkat lunak adalah sebagai berikut:

- a. Semua pengujian harus dapat di urutkan sampai kepada spesifikasi kebutuhan perangkat lunak.
- b. Pengujian harus dimulai dari lingkup yang kecil sampai lingkup yang besar.
- c. Pengujian yang mendalam tidak mungkin dilakukan karena tidak mungkin mengeksekusi semua jalur permutasi.
- d. Supaya efektif (memiliki probabilitas yang tinggi dalam menemukan kesalahan), pengujian harus dilakukan oleh pihak lain yang independen.

### **2.16.3 Pengujian Black Box**

Metode pengujian Black Box merupakan metode yang menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksud untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Raka, 2016). Metode pengujian black box adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat satu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya. Fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya. (hanya mengetahui input dan output) (Desi, 2014). Pengujian black box berusaha menemukan kesalahan dalam kategori sebagai berikut:

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan interface.
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan kinerja.
- e. Insialisasi dan kesalahan terminasi.

Pengujian black box cenderung diaplikasikan selama tahap akhir pengujian, karena pengujian black box memperhatikan struktur control, maka perhatian berfokus pada domain informasi, pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut ini:

- a. Bagaimana validasi fungsional diuji?
- b. Kesalahan input apa yang akan membuat test case menjadi baik?

- c. Apakah sistem sangat sensitive terhadap harga input tertentu?
- d. Bagaimana batasan dari suatu data diisolasi?
- e. Kecepatan apa dan volume data apa saja dapat ditolerir oleh sistem?
- f. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem?

Dengan mengaplikasikan teknik *black box*, maka menarik serangkaian *test case* yang memenuhi kriteria berikut ini:

- a. *Test case* yang mengurangi, dengan harga lebih dari satu, jumlah test case tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggung jawabkan.
- b. *Test case* yang memberi tahu sesuatu mengenai kehadiran atau ketidakhadiran kelas kesalahan, dari pada memberitahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada.

## **BAB III**

### **ANALISA DAN PERANCANGAN**

#### **3.1 Analisa Sistem**

Analisa sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian dalam komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan- kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Ramadhani & Siregar, 2022)

Sebelum dilakukan tahap perancangan sebuah sistem, perlu dilakukan analisa sistem yang akan dibangun. Analisis sistem merupakan istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Tahap ini bertujuan untuk memberikan gambaran yang jelas terhadap sistem yang akan dibangun. Tahap ini menjabarkan kebutuhan-kebutuhan yang berguna untuk perancangan sistem agar sistem yang dibangun sesuai dengan masalah yang akan diselesaikan.

##### **3.1.1 Analisa Masalah**

Tahap pertama yang dilakukan adalah mendefinisikan permasalahan yang terjadi. Permasalahan yang dihadapi dalam penelitian ini adalah bagaimana membantu anak untuk senang terhadap pelajaran matematika. Dalam tahap ini analisis tentang masalah sistem, yaitu kurangnya minat anak karena tidak adanya hiburan pada buku pelajaran yang hanya berupa tulisan yang membuat pembelajaran itu sendiri terkesan membosankan dan monoton.

Dari permasalahan tersebut maka dibuatlah media pembelajaran *game* edukasi matematika berbasis *mobile* yang dikemas kedalam bentuk visualisasi animasi yang diharapkan dapat mengembalikan semangat anak dalam belajar matematika.



### 3.1.2 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem merupakan beberapa kebutuhan bahan dalam sistem yang akan dipergunakan untuk menambah dan membantu jalannya proses pembuatan suatu objek. Dibagian ini akan dibagi menjadi dua bagian, yaitu:

#### 1. Analisa Kebutuhan Fungsional

Analisis kebutuhan fungsional berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional juga berisi informasi apa saja yang harus ada dan dihasilkan oleh sistem yaitu:

- a. Menampilkan materi matematika dalam bentuk animasi secara acak atau *random*.
- b. Menampilkan *quiz* yang berisi pilihan ganda tentang penjumlahan, pengurangan, perkalian dan pembagian.
- c. Permainan yang berfungsi untuk mengasah logika pemain.

#### 2. Analisa Kebutuhan Non-Fungsional

Analisa kebutuhan non-fungsional adalah tipe kebutuhan yang berisi property yang dimiliki sistem. Kebutuhan ini dapat dibagi menjadi dua bagian:

##### A. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah:

- a. Laptop Intel Core i5 8<sup>th</sup> Gen
- b. SSD M.2 128GB
- c. RAM 12GB
- d. Smartphone

##### B. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah:

- a. Sistem Operasi Windows 10
- b. Unity 3D
- c. Visual Paradigm
- d. Adobe XD
- e. Adobe Photoshop
- f. Visual Studio Code

### 3.2 Perancangan Aplikasi

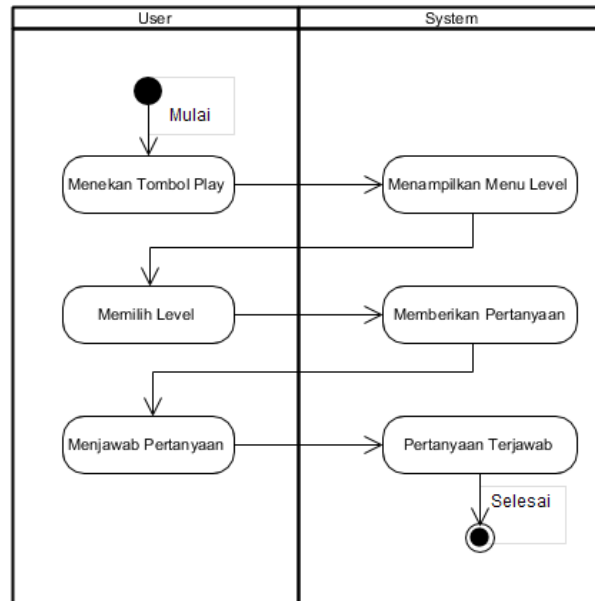
Perancangan aplikasi merupakan tahap selanjutnya setelah analisa sistem mendapatkan gambaran dengan jelas tentang apa saja yang dikerjakan. Maka dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut. Pada perancangan sistem ini menggunakan UML (*Unified Modeling Language*) yang meliputi *use case* diagram, *activity* diagram, dan *sequence* diagram. Dalam perancangan aplikasi ini juga terdapat user *interface* (antarmuka) yang merupakan langkah awal dalam membuat tampilan Aplikasi *Game Solve The Math* Berbasis Android.

#### 3.2.1 Gambaran Umum Sistem Usulan

Gambaran umum dari sistem yang diusulkan untuk memberikan gambaran secara umum kepada pengguna (*user*) mengenai sistem aplikasi berbasis android. Beberapa hal yang menjadi batasan masalah yang akan diberikan solusi atau alternatif dengan maksud menjelaskan tentang kebutuhan-kebutuhan yang harus dipenuhi oleh sistem *game* edukasi yang akan dirancang, berdasarkan hasil analisa, maka dibuat suatu kebutuhan dalam perancangan.

Perancangan sistem perangkat lunak dideskripsikan dengan model analisis menggunakan diagram *use case* serta dilanjutkan dengan model *design*. Analisis digunakan untuk pemetaan awal mengenai perilaku yang diisyaratkan sistem aplikasi kedalam elemen-elemen permodelan. Untuk melengkapi dokumentasi perancangan digunakan *Visual Paradigm* yang mendukung perancangan berbasis UML.

Implementasi menggambarkan bagaimana suatu sistem dibentuk. Pada tahap perancangan aplikasi android dirancang dengan tujuan sebagai alat komunikasi antara pengguna (*user*) dengan pembuat program guna mendapatkan sistem aplikasi yang sesuai dengan apa yang dibutuhkan.



**Gambar 3. 1 Gambaran Umum Sistem Usulan**

### 3.2.2 Prosedur Sistem Yang Diusulkan

Prosedur sistem yang diusulkan adalah:

- Pengguna dapat membuka aplikasi *game* edukasi untuk memulai permainan.
- Setelah itu pengguna akan masuk ke halaman utama dan menekan tombol bermain.
- Pengguna harus memilih level yang bisa dimainkan, didalam permainan, pengguna akan menjawab beberapa pertanyaan yang ada untuk menyelesaikan permainan.

### 3.3 Perancangan Sistem

Rancangan sistem secara umum dilakukan dengan maksud untuk memberikan gambaran umum tentang sistem yang baru atau sistem yang akan diusulkan. Rancangan ini mendefinisikan komponen-komponen sistem informasi yang dirancang secara rinci.

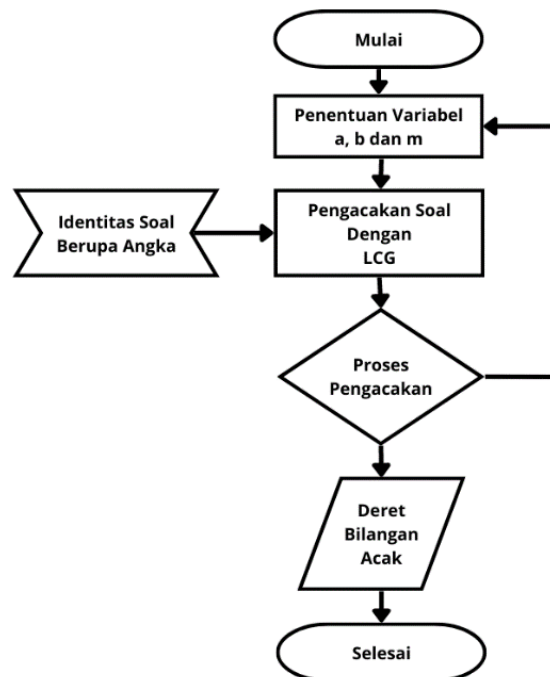
### 3.4 Perancangan Soal

Pada game ini metode *Linear Congruential Generator* diterapkan pada pengacakan soal, soal diterapkan di setiap level. Dengan soal yang diacak dengan metode *Linear Congruential Generator* sehingga memiliki urutan yang berbeda dengan sebelumnya, hal ini dilakukan agar soal tidak keluar berulang sehingga user tidak jenuh apabila bermain dikarenakan mendapatkan soal yang sama.

### 3.5 Metode Linear Congruential Generator

Metode *Linear Congruential Generator* (LCG) digunakan untuk menghasilkan bilangan acak yang digunakan dalam permainan *Solve The Math* ini. Bilangan acak yang dihasilkan oleh *Linear Congruential Generator* (LCG) ini akan digunakan untuk menentukan posisi dari bilangan 1 sampai n pada setiap soal.

#### 3.5.1 Flowchart Metode Linear Congruential Generator



**Gambar 3. 2** *Flowchart Metode Linear Congruential Generator*

Berdasarkan **Gambar 3.2** terlihat bahwa setelah penentuan identitas soal yang akan digunakan dan penentuan variabel a,b dan m maka dapat dilakukan proses pengacakan soal dengan LCG. Sistem akan meminta penentuan ulang

variabel  $a, b$  dan  $m$  apabila terjadi kegagalan selama proses pengacakan soal. Hasil dari pengacakan soal adalah deret bilangan yang merupakan identitas masing-masing soal.

### 3.5.2 Implementasi Linear Congruential Generator

Metode *Linear Congruential Generator* diimplementasikan dengan alur sebagai berikut:

1. Memulai permainan.
2. Inisialisasi konten yang ada dalam *game*.
3. Mengacak pertanyaan menggunakan *Linear Congruential Generator*.
4. Menampilkan pertanyaan.



**Gambar 3. 3 Diagram Impelementasi *Linear Congruential Generator***

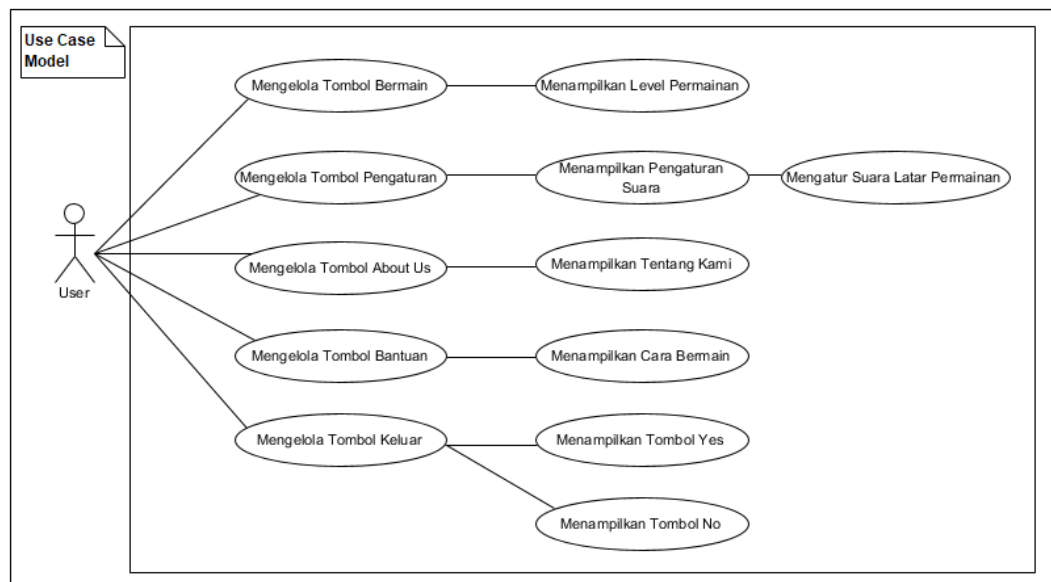
### 3.6 Perancangan Aplikasi

Perancangan adalah suatu kegiatan yang memiliki tujuan untuk mendesign sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik (Ladjamudin, 2005).

Berdasarkan definisi di atas, penulis menarik kesimpulan bahwa perancangan merupakan suatu pola yang dibuat untuk mengatasi masalah yang dihadapi perusahaan atau organisasi setelah melakukan analisis terlebih dahulu.

### 3.6.1 Use Case Diagram

*Use Case* diagram digunakan untuk menjelaskan apa yang akan dilakukan oleh sistem serta aktor yang berhubungan dengan proses yang ada pada sistem. Pada perancangan *use case* diagram ini akan menjelaskan hal-hal yang dapat dilakukan *user* atau pengguna Aplikasi *Game* Edukasi Berbasis *Mobile*.



**Gambar 3. 4 Use Case Diagram**

Gambar 3.4 adalah gambar *use case* diagram aplikasi *game* edukasi yang dapat melakukan beberapa hal seperti pada gambar *use case diagram* diatas. Untuk penjelasan gambar tersebut dapat dilihat pada tabel sebagai berikut:

**Table 3. 1 Tabel Use Case Diagram**

Simbol	Deskripsi
User	Pengguna Aplikasi. Aplikasi ini dapat digunakan oleh siapa saja

Menampilkan Tombol Bermain	Merupakan proses pemilihan Tombol Bermain untuk menampilkan halaman pilih level
Melihat Level Permainan	Merupakan proses mengetahui banyaknya jumlah level pada permainan
Menampilkan Tombol Pengaturan	Merupakan proses pemilihan Tombol Pengaturan untuk mengatur suara latar dan efek suara dari permainan
Mengatur Suara Latar	Merupakan proses mengatur volume suara latar permainan
Menampilkan Tombol Tentang Kami	Merupakan proses pemilihan tombol Tentang Kami jika ingin melihat biodata perancang aplikasi
Melihat Biodata Perancang	Merupakan proses mengetahui kepemilikan suatu aplikasi
Menampilkan Tombol Keluar	Merupakan proses pemilihan tombol keluar jika ingin keluar dari permainan
Menampilkan Tombol Iya	Merupakan proses pemilihan tombol Iya untuk mengkonfirmasi jika ingin keluar dari permainan
Menampilkan Tombol Tidak	Merupakan proses pemilihan tombol Tidak untuk mengkonfirmasi jika tidak ingin keluar dari permainan

Deskripsi *Use Case Diagram* sebagai berikut:

- a. Mengelola Halaman Utama

Pada halaman ini pengguna akan ditampilkan menu utama dari Aplikasi *Game* Edukasi berbasis Mobile. Di halaman utama ini akan disajikan beberapa tombol, yaitu tombol: Tentang Aplikasi, Bermain, Pengaturan dan Keluar.

b. Mengelola Tombol Bermain

Pada menu tombol Bermain, pengguna akan disajikan pilihan level yang dapat dimainkan. Setelah memilih level pengguna akan disajikan *game trivia*. *Game* ini bertujuan untuk melatih logika pengguna untuk menyelesaikan beberapa pertanyaan di setiap levelnya untuk melanjutkan ke level berikutnya.

c. Mengelola Tombol Pengaturan

Pada tombol Pengaturan pengguna dapat mematikan atau menyalakan music yang ada pada game tersebut.

d. Mengelola Tentang Kami

Pada tombol Tentang Kami pengguna akan mendapatkan informasi tentang pembuatan Aplikasi *Game* Edukasi *Solve The Math*.

e. Mengelola Tombol Keluar

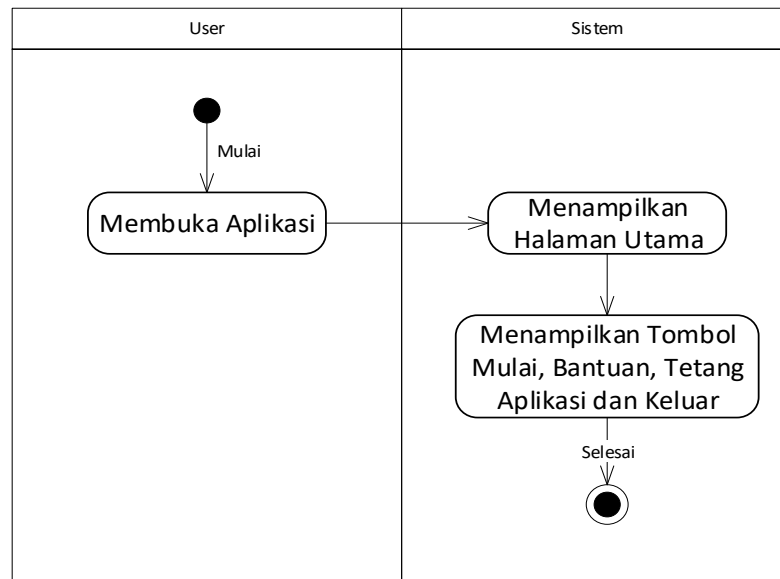
Tombol Keluar digunakan untuk keluar dari aplikasi. Namun pengguna juga dapat keluar dari aplikasi ini dengan menggunakan tombol *back* pada *smartphone* yang digunakan.

### 3.6.2 Activity Diagram

*Activity* Diagram mempunyai manfaat dalam membantu memahami suatu proses secara keseluruhan apabila *activity* diagram ini dibuat terlebih dahulu sebelum memodelkan suatu proses. Dalam artian yang lain *activity* diagram dapat berupa gambaran suatu proses, dipakai pada bisnis modeling untuk memperlihatkan urutan aktifitas proses bisnis, memiliki struktur diagram yang mirip *flowchart* atau data flow diagram pada perancangan terstruktur. *Activity* diagram ini bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan *activity* diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram. Dan berikut adalah tampilan dari *activity* diagram yang ada pada Aplikasi *Game* Edukasi.



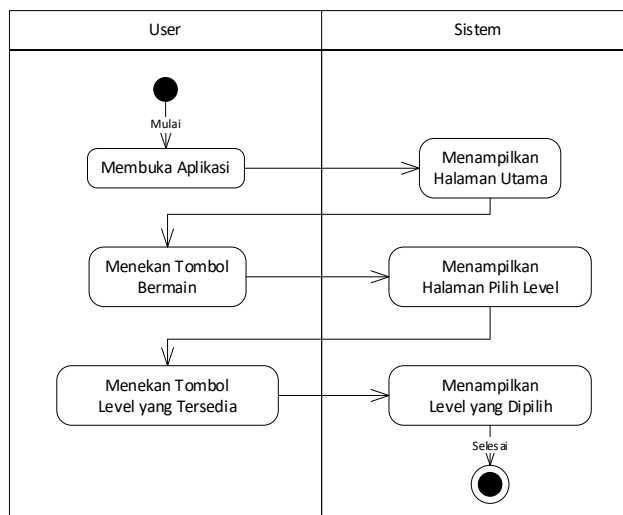
a. Diagram Halaman Utama



**Gambar 3. 5 Activity Diagram Halaman Utama**

Pada Gambar 3.5 merupakan Activity Diagram dari Halaman Utama pada sistem. Gambar diatas menggambarkan tampilan utama yang muncul saat pengguna membuka Aplikasi *Game Edukasi Solve The Math* Berbasis Android.

b. Diagram Tombol Bermain

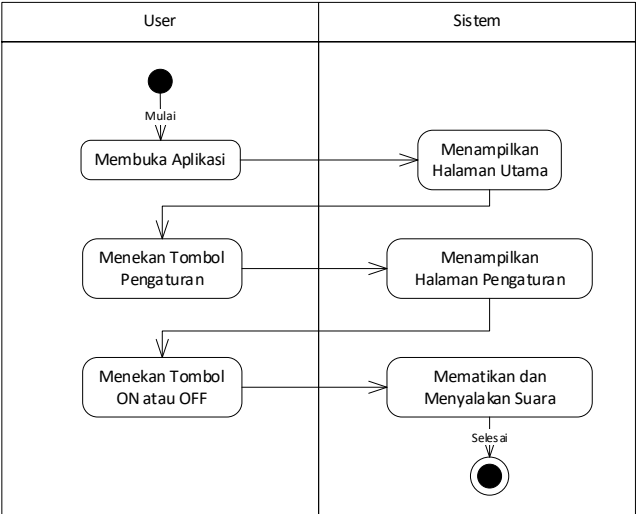


**Gambar 3. 6 Activity Diagram Tombol Bermain**

Pada Gambar 3.6 merupakan Activity Diagram dari Tombol Bermain pada sistem. Gambar diatas menggambarkan bagaimana sistem membaca perintah

saat pengguna menekan Tombol Bermain. Saat itu sistem akan menampilkan semua level permainan.

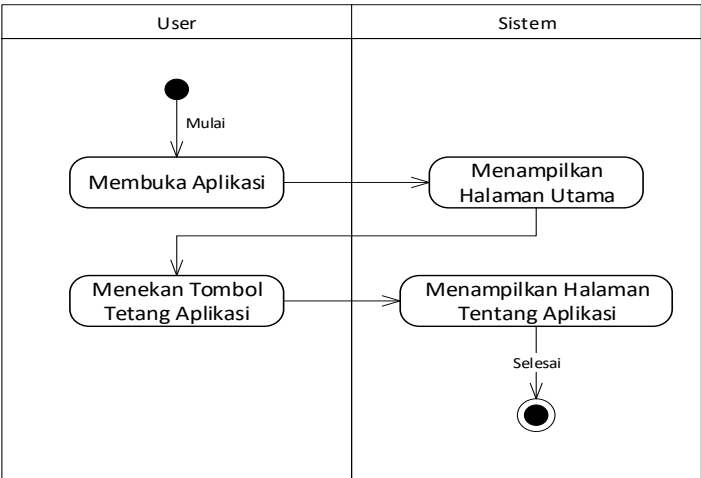
c. Diagram Tombol Pengaturan



**Gambar 3. 7 Activity Diagram Tombol Pengaturan**

Pada Gambar 3.7 merupakan *Activity* Diagram dari Tombol Pengaturan pada sistem. Gambar diatas menjelaskan bagaimana sistem membaca perintah saat pengguna menekan Tombol Pengaturan. Saat itu sistem akan menampilkan pengaturan suara yang dapat dihidupkan dan dimatikan.

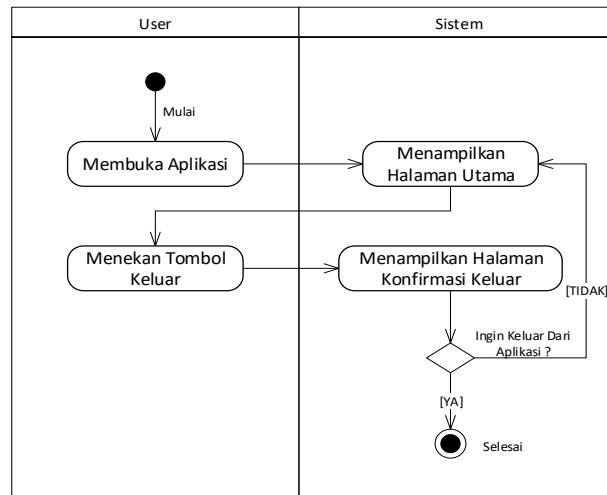
d. Diagram Tombol About Us



**Gambar 3. 8 Activity Diagram About Us**

Pada Gambar 3.8 merupakan *Activity Diagram* dari Tombol Tentang Kami dalam *game*. Dari gambar diatas menjelaskan tentang pengguna dapat melihat tentang kami.

e. Diagram Tombol Keluar



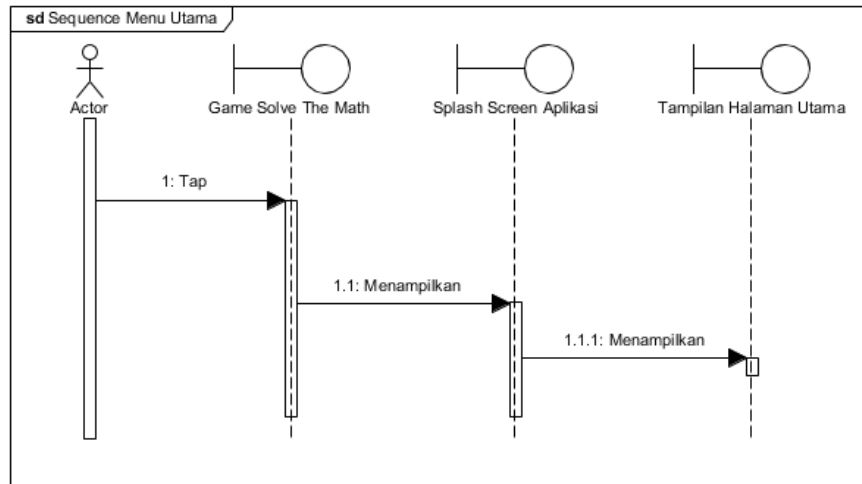
**Gambar 3. 9 Activity Diagram Tombol Keluar**

Pada Gambar 3.9 merupakan Activity diagram dari tombol keluar dalam *game*. Dari gambar diatas menggambarkan bagaimana sistem membaca perintah saat pengguna menekan tombol keluar. Saat itu sistem akan menampilkan konfirmasi sebelum keluar dari aplikasi.

### 3.6.3 Sequence Diagram

*Sequence Diagram* menggambarkan objek yang berpartisipasi dalam sebuah *use case* dan pesan yang terdapat di antara mereka dari waktu ke waktu untuk suatu *use case*, *sequence diagram* juga dipakai untuk memodelkan deskripsi tentang sistem yang ada pada sebuah atau beberapa *use case* pada *use case diagram*, yang menggambarkan hubungan antara aktor dan *use case*. Berikut ini adalah *sequence diagram* yang terdapat pada aplikasi.

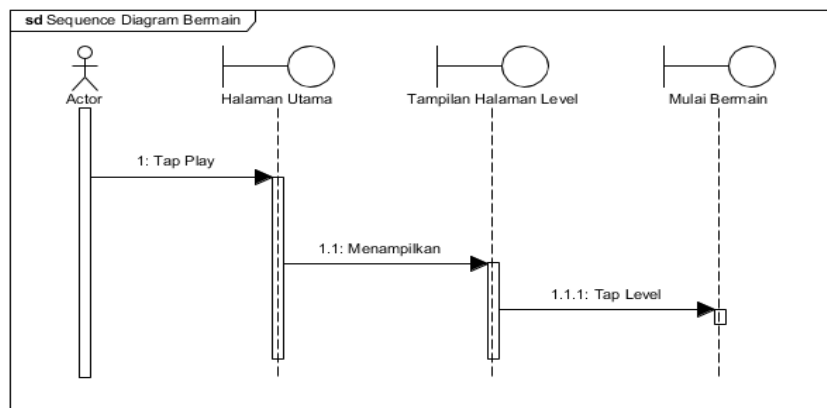
a. *Sequence Diagram Halaman Utama*



**Gambar 3. 10 *Sequence Diagram* Halaman Utama**

Pada *Sequence Diagram* Mulai tersebut dapat terlihat bahwa pada saat pengguna membuka aplikasi *game solve the math* berbasis *mobile* sistem akan merespon dengan menampilkan halaman utama dari aplikasi tersebut. Kemudian pada halaman utama ini terdapat lima tombol yang ditampilkan oleh sistem, yaitu tombol bermain, pengaturan, bantuan, tentang aplikasi dan keluar.

b. *Sequence Diagram Bermain*

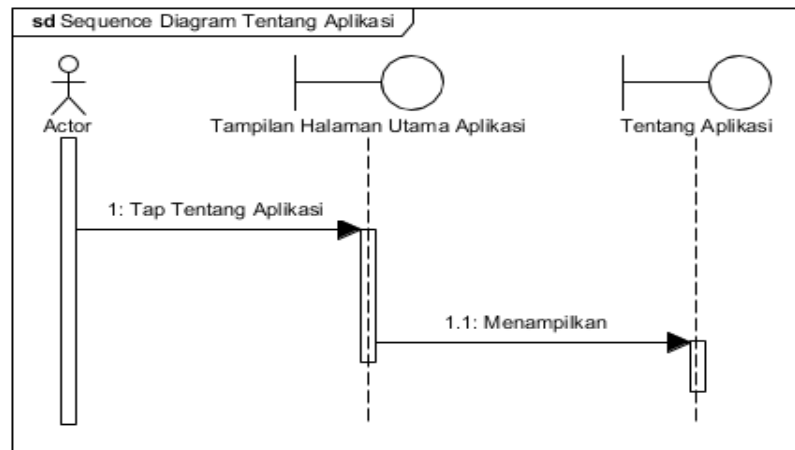


**Gambar 3. 11 *Sequence Diagram* Bermain**

Pada *Sequence Diagram* Bermain tersebut dapat terlihat bahwa pada saat pengguna membuka aplikasi sistem akan merespon dengan menampilkan halaman utama dari aplikasi tersebut. Kemudian pada halaman utama ini terdapat lima tombol yang ditampilkan oleh sistem. Setelah pengguna menekan tombol bermain,

maka sistem akan menampilkan halaman memilih level, yang dimana pengguna hanya bisa memainkan level yang sudah terbuka saja.

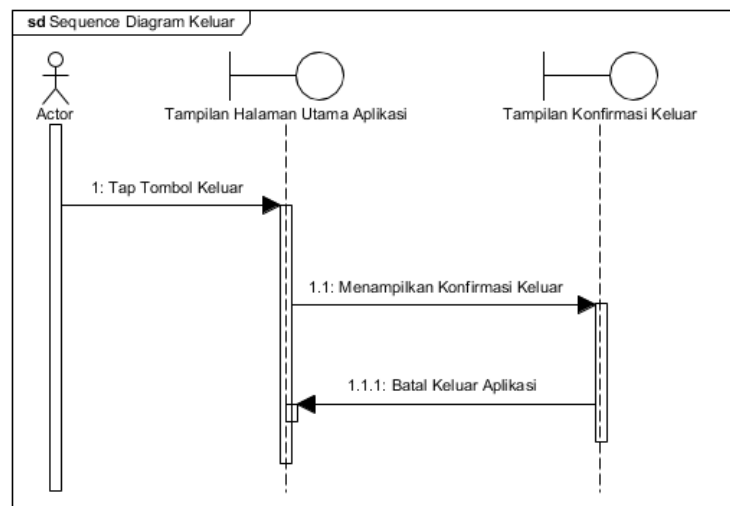
c. *Sequence Diagram Tentang Aplikasi*



**Gambar 3. 12 *Sequence Diagram Tentang Aplikasi***

*Sequence Diagram Tentang Aplikasi* di atas menggambarkan bahwa pada saat pengguna memilih tentang aplikasi maka sistem akan menampilkan informasi versi aplikasi dan nama pembuat dari aplikasi tersebut.

d. *Sequence Diagram Keluar*

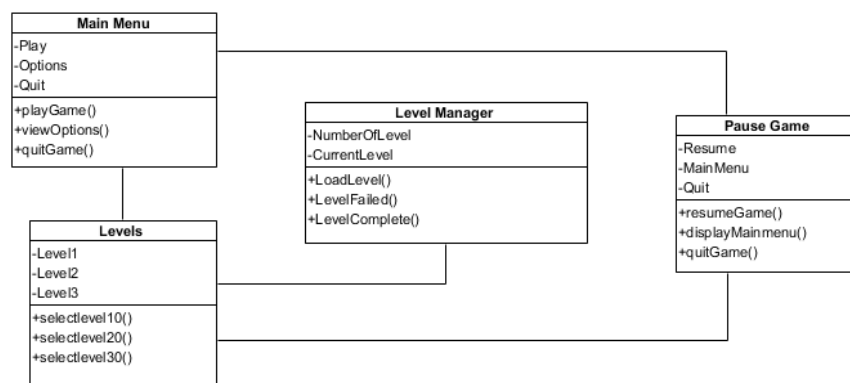


**Gambar 3. 13 *Sequence Diagram Keluar***

*Sequence* Diagram Keluar tersebut dapat terlihat bahwa pengguna harus melakukan konfirmasi terlebih dahulu untuk dapat keluar dari aplikasi.

### 3.6.4 Class Diagram

*Class* Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan desain berorientasi objek. *Class* menggambarkan keadaan (attribut dan property) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi).



**Gambar 3. 14 Class Diagram**

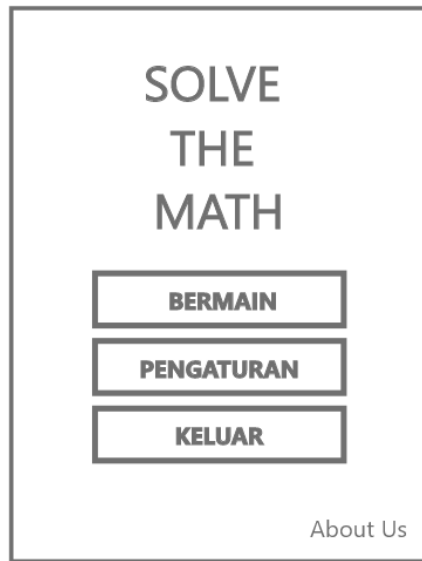
### 3.6.5 Perancangan Layar

Perancangan layar pada bab ini dijelaskan dalam dua rancangan diantaranya, rancangan halaman utama, rancangan tentang kami, rancangan halaman bantuan, rancangan halaman permainan, rancangan halaman pengaturan.

#### 3.6.5.1 Rancangan Halaman Utama

Rancangan Halaman Utama dapat dilihat pada gambar 3.15 Pada tampilan halaman awal terdapat berbagai tombol seperti Bermain, Pengaturan, Keluar, dan *About Us*. Pada background terdapat animasi bergerak agar lebih menarik minat anak.

Rancangan Halaman Utama dapat dilihat pada gambar berikut:



**Gambar 3. 15 Menu Utama Aplikasi**

#### **3.6.5.2 Rancangan Halaman Tentang Kami**

Rancangan Halaman Tentang Kami dapat dilihat pada gambar 3.16 pada halaman tentang kami terdapat Nama Perancang aplikasi dan Keterangan tentang Aplikasi.

Rancangan Halaman Tentang Kami dapat dilihat pada gambar berikut:

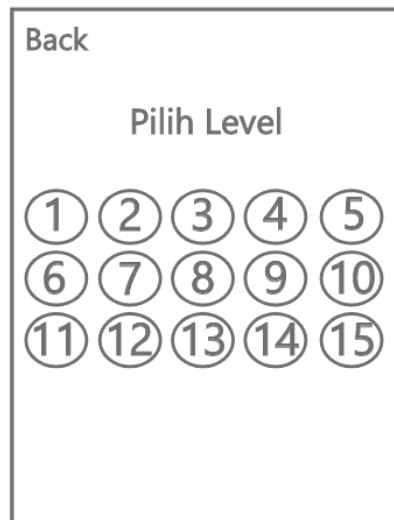


**Gambar 3. 16 Menu *About Us***

#### **3.6.5.3 Rancangan Halaman Pilih Level**

Halaman Pilih Level dapat dilihat pada gambar 3.17 Pada Halaman Pilih Level terdapat jumlah level yang dapat dimainkan oleh pemain, yang dimana para

pemain harus menyelesaikan level yang tersedia untuk bisa lanjut ke level berikutnya.

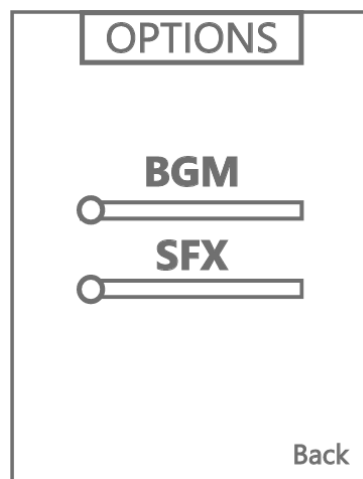


**Gambar 3. 17 Rancangan Halaman Pilih Level**

#### **3.6.5.4 Rancangan Halaman Pengaturan**

Halamn Pengaturan dapat dilihat pada gambar 3.18 Pada Halaman Pengaturan terdapat dua tombol untuk mengatur suara hidup atau mati pada permainan.

Rancangan Halaman Pengaturan dapat dilihat pada gambar berikut:

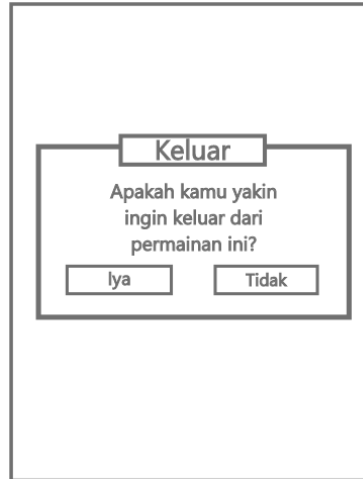


**Gambar 3. 18 Rancangan Halaman Pengaturan**



#### 3.6.5.5 Rancangan Tampilan Konfirmasi Keluar

Tampilan pada Konfirmasi Keluar dapat dilihat pada Gambar 3.19 Pada Tampilan Konfirmasi Keluar terdapat dua tombol untuk menentukan pilihan apakah ingin keluar dari aplikasi atau tidak.



**Gambar 3. 19 Rancangan Tampilan Konfirmasi Keluar**

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi

Pokok pembahasan pada bab ini adalah implementasi dan pengujian terhadap Aplikasi *Game* Edukasi *Solve The Math* Berbasis Android. Tahapan implementasi dilakukan setelah proses analisa dan perancangan selesai dilakukan yang kemudian akan diimplementasikan kedalam bahasa pemrograman. Kemudian setelah implementasi maka selanjutnya akan dilakukan pengujian terhadap aplikasi ini. Aplikasi yang telah dibangun akan diimplementasikan untuk dapat diketahui apakah sistem tersebut dapat berjalan sesuai dengan tujuannya atau tidak.

##### 4.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan pada perancangan program aplikasi *game* edukasi *Solve The Math* berbasis *mobile* ini, yaitu:

**Table 4. 1 Perangkat Keras Yang Digunakan**

No	Perangkat Keras	Spesifikasi
<b>LAPTOP</b>		
1.	Prosesor	Prosesor Intel Core i5-8250U 8 <sup>th</sup> Gen
2.	Monitor	14" (inch)
3.	RAM	12 GB
4.	Graphics	NVIDIA® GeForce® MX150
5.	Hard Disk Drive	1 TB
6.	Solid State Drive	256 GB
7.	Operating System	Windows 10

<b>MOBILE</b>		
1.	Layar	6.5'' (inch)
2.	CPU	MediaTek MT6765 Helio P35 (12nm)
3.	Graphics	PowerVR GE8320
4.	Kecepatan CPU	Octa-core (4x2.35 GHz Cortex-A53 & 4x1.8 GHz Cortex-A53)
5.	RAM	4 GB
6.	Storage	64 GB
7.	Android Version	Android 12

#### 4.1.2 Spesifikasi Perangkat Lunak

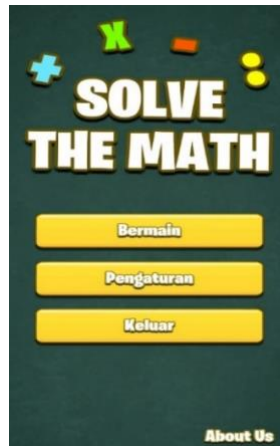
Perangkat lunak yang digunakan pada perancangan program aplikasi *game* edukasi *Solve The Math* berbasis *mobile* ini, yaitu:

1. Sistem Operasi *Windows 10*
2. *Unity 3D 2020 Ver. 2020.3.30f1*
3. *Visual Studio Code*
4. *Adobe Photoshop CC*
5. *Microsoft Visio*
6. *Adobe XD*

#### 4.1.3 Implementasi Antarmuka

Pada tahap implementasi antarmuka atau *user interface* menggambarkan tampilan dari aplikasi yang dibangun yaitu *Game Edukasi Solve The Math*. Implementasi antarmuka menggunakan hasil *screen capture* untuk setiap menu yang ada pada aplikasi. Berikut adalah implementasi antarmuka atau *user interface*.

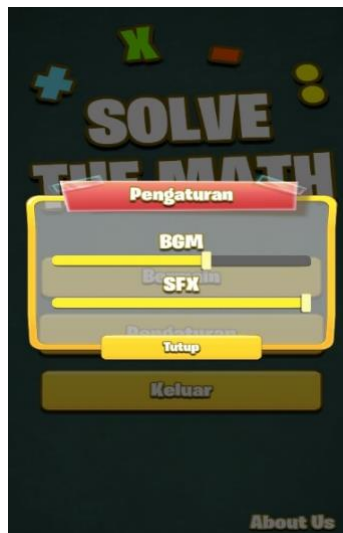
a. Tampilan Halaman Utama



**Gambar 4. 1 Tampilan Halaman Utama**

Pada Gambar 4.1 ditunjukkan tampilan halaman utama. Tampilan halaman utama ini berfungsi untuk *Menu View* atau tampilan awal dari berbagai tombol seperti Bermain, Pengaturan, Keluar dan *About Us*.

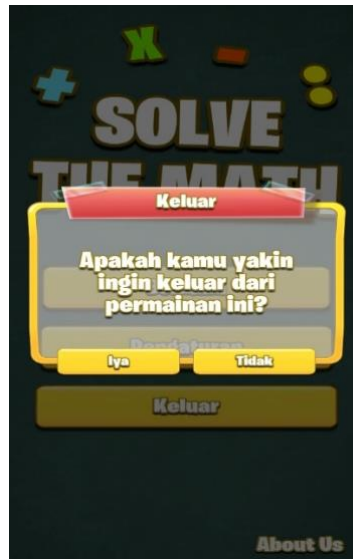
b. Tampilan Halaman Pengaturan



**Gambar 4. 2 Tampilan Halaman Pengaturan**

Pada Gambar 4.2 ditunjukkan tampilan halaman pengaturan. Tampilan halaman pengaturan ini berfungsi untuk menampilkan pengaturan suara latar dan efek suara pada permainan.

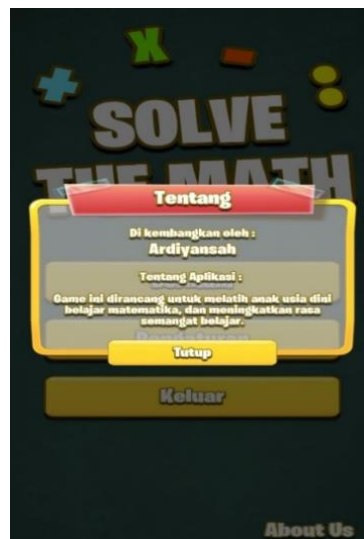
c. Tampilan Halaman Keluar



**Gambar 4. 3 Tampilan Halaman Keluar**

Pada Gambar 4.3 ditunjukkan tampilan halaman keluar. Tampilan menu keluar ini berfungsi untuk keluar dari game dan terdapat dua buah tombol pilihan Iya atau Tidak.

d. Tampilan About Us



**Gambar 4. 4 Tampilan About Us**

Pada Gambar 4.4 ditunjukkan tampilan pada halaman *About Us*. Tampilan menu *About Us* berfungsi untuk menampilkan informasi nama perancang aplikasi dan tujuan aplikasi dirancang.

e. Tampilan Pilih Level



**Gambar 4. 5 Tampilan Pilih Level**

Pada Gambar 4.5 ditunjukkan Tampilan Pilih Level Permainan. Tampilan Pilih Level Permainan ini berfungsi untuk menampilkan level permainan yang dapat dimainkan.

f. Tampilan Pertanyaan Penjumlahan



**Gambar 4. 6 Tampilan Pertanyaan Penjumlahan**

Pada Gambar 4.6 ditunjukkan tampilan pertanyaan penjumlahan. Pertanyaan setiap level terdiri dari 5 buah pertanyaan yang dimana dapat ditampilkan secara acak dan diberikan waktu untuk menjawab setiap soal yang diberikan.

g. Tampilan Pertanyaan Pengurangan



**Gambar 4. 7 Tampilan Pertanyaan Pengurangan**

Pada Gambar 4.7 ditunjukkan tampilan pertanyaan pengurangan. Pertanyaan setiap level terdiri dari 5 buah pertanyaan yang dimana dapat ditampilkan secara acak dan diberikan waktu untuk menjawab setiap soal yang diberikan.

h. Tampilan Pertanyaan Perkalian



**Gambar 4. 8 Tampilan Pertanyaan Perkalian**

Pada Gambar 4.8 ditunjukkan tampilan pertanyaan perkalian. Pertanyaan setiap level terdiri dari 5 buah pertanyaan yang dimana dapat ditampilkan secara acak dan diberikan waktu untuk menjawab setiap soal yang diberikan.

i. Tampilan Pertanyaan Pembagian



**Gambar 4. 9 Tampilan Pertanyaan Pembagian**

Pada Gambar 4.9 ditunjukkan tampilan pertanyaan pembagian. Pertanyaan setiap level terdiri dari 5 buah pertanyaan yang dimana dapat ditampilkan secara acak dan diberikan waktu untuk menjawab setiap soal yang diberikan.

## **4.2 Pengujian Sistem**

Pengujian sistem berarti proses untuk memeriksa apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai dengan standar tertentu. Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menentukan kesalahan.

### **4.2.1 Pengujian *Black Box***

Proses berikutnya yaitu sebelum program diterapkan, maka harus dipastikan terlebih dahulu bahwa program sudah bebas dari kesalahan-kesalahan, maka dari



pada itu program harus dilakukan uji coba untuk menemukan setiap kesalahan-kesalahan yang mungkin dapat terjadi, pada pengujian aplikasi ini metode pengujian yang digunakan adalah metode pengujian *Black Box*.

Pengujian *Black Box* adalah suatu pengujian program yang menguji pada fungsionalitas dari program tersebut. Pengujian dengan menggunakan metode *Black Box* ini dilakukan dengan cara memberikan sejumlah input atau masukan pada program aplikasi tersebut yang kemudian diproses sesuai dengan kebutuhan fungsionalnya yang nantinya akan dilihat apakah program aplikasi menghasilkan output atau keluaran yang diinginkan dan sesuai dengan fungsi dari program aplikasi tersebut. Dan apabila dari input yang diberikan proses menghasilkan output yang sudah sesuai dengan kebutuhan fungsionalnya, maka program aplikasi tersebut sudah benar, akan tetapi jika output yang dihasilkan masih tidak sesuai dengan kebutuhan fungsionalnya, maka kemungkinan masih terdapat kesalahan pada program aplikasi tersebut.

Pengujian dengan menggunakan metode *Black Box* dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan secara berulang-ulang. Apabila pada proses pengujian ditemukan kesalahan, maka akan dilakukan *tracking* dan perbaikan untuk memperbaiki kesalahan-kesalahan yang terjadi. Dan jika sudah selesai, perbaikan dilakukan secara berulang-ulang sehingga dapat diperoleh hasil yang terbaik.

Metode pengujian *Black Box* berfokus pada persyaratan fungsional dari suatu *software*, maka dengan demikian pengujian *Black Box* memungkinkan perekrut *software* atau perangkat lunak dapat mengungkap kesalahan-kesalahan dalam kategori sebagai berikut:

- a. Fungsi-fungsi yang tidak benar atau salah
- b. Kesalahan *interface*
- c. Kesalahan kinerja *software*
- d. Inisialisasi dan kesalahan terminasi

Kemudian untuk proses pengujian yang dilakukan, pengujian berdasarkan dari tampilan yang ada pada sistem yang dibuat yang nantinya akan difungsikan oleh pengguna, pengujian tersebut dapat digambarkan dalam tabel-tabel berikut ini:

**Table 4.2 Table Rencana Pengujian**


Item Uji	Jenis Pengujian
Mengelola Halaman Utama	Black Box
Mengelola Tombol Bermain	Black Box
Mengelola Tombol Pengaturan	Black Box
Mengelola Tombol <i>About Us</i>	Black Box
Mengelola Tombol <i>Pause</i>	Black Box



#### 4.2.2 Kasus dan Hasil Pengujian

Berdasarkan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut:

##### a. Pengujian Halaman Utama

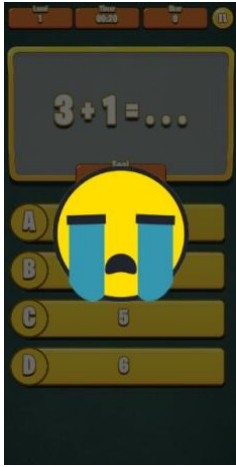

**Table 4. 2 Pengujian Halaman Utama**

Interaksi yang dimasukkan	Yang diharapkan	Pengamatan	Kesimpulan
Tap pada tombol Bermain.	Menampilkan halaman level permainan.  Hasil Pengujian: 	Sesuai Harapan	(✓) diterima ( ) ditolak

<p>Tap pada tombol Pengaturan</p>	<p>Dapat menampilkan halaman Pengaturan</p> <p>Hasil Pengujian:</p> 	<p>Sesuai Harapan</p>	<p>(✓) diterima ( ) ditolak</p>
<p>Tap pada tombol <i>About Us</i></p>	<p>Dapat menampilkan halaman <i>About Us</i></p> <p>Hasil Pengujian:</p> 	<p>Sesuai Harapan</p>	<p>(✓) diterima ( ) ditolak</p>



b. Halaman Pertanyaan


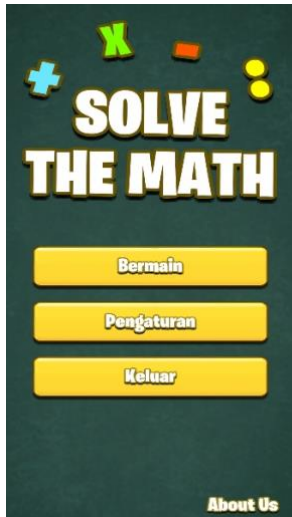
**Table 4. 3 Pengujian Halaman Pertanyaan**

<b>Interaksi yang dimasukkan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Pertanyaan salah	<p>Jika salah menjawab pertanyaan maka akan menampilkan ikon.</p> <p>Hasil Pengujian:</p> 	Sesuai Harapan	(✓) diterima ( ) ditolak
Pertanyaan benar	<p>Jika benar menjawab pertanyaan maka akan menampilkan ikon.</p> 	Sesuai Harapan	(✓) diterima ( ) ditolak

c. Pengujian Tombol Pause



**Table 4. 4 Pengujian Tombol Pause**

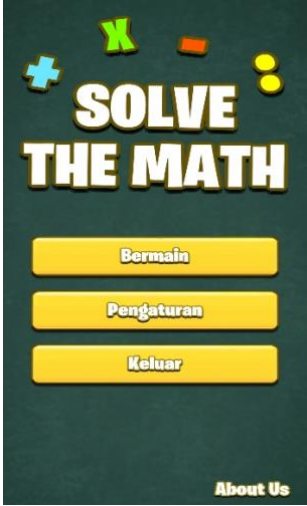
<b>Interaksi yang dimasukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Tap pada tombol Pause	<p>Dapat menghentikan permainan sejenak</p> <p>Hasil pengujian:</p> 	Sesuai Harapan	<p>(✓) diterima</p> <p>( ) ditolak</p>
Tap pada tombol Resume	<p>Dapat melanjutkan permainan</p> <p>Hasil Pengujian:</p> 	Sesuai Harapan	<p>(✓) diterima</p> <p>( ) ditolak</p>

<p>Tap pada tombol Restart</p>	<p>Dapat mengulang permainan dan memulainya dari awal</p> <p>Hasil Pengujian:</p> 	<p>Sesuai Harapan</p>	<p>(✓) diterima ( ) ditolak</p>
<p>Tap pada tombol Main Menu</p>	<p>Dapat menampilkan halaman utama</p> <p>Hasil Pengujian:</p> 	<p>Sesuai Harapan</p>	<p>(✓) diterima ( ) ditolak</p>

d. Pengujian Tombol Pengaturan

**Table 4. 5 Pengujian Tombol Pengaturan**

<b>Interaksi yang dimasukkan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Slide pada Pengaturan	<p>Dapat mengatur suara latar pada permainan</p> <p>Hasil Pengujian:</p> 	Sesuai Harapan	<p>(✓) diterima</p> <p>( ) ditolak</p>
Slide pada pengaturan suara efek	<p>Dapat mengatur efek suara background musik pada permainan</p> <p>Hasil Pengujian:</p> 	Sesuai Harapan	<p>(✓) diterima</p> <p>( ) ditolak</p>

Tap pada tombol kembali	<p>Dapat Kembali ke halaman utama</p> <p>Hasil Pengujian:</p> 	Sesuai Harapan	<p>(✓) diterima</p> <p>( ) ditolak</p>
-------------------------	---	----------------	--



## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dimulai dari perancangan aplikasi sampai dengan pengujian dan pengimplementasian Aplikasi *Game* Edukasi *Solve The Math* Berbasis Android ini, dapat disimpulkan:

1. Aplikasi dirancang dengan menggunakan metode RAD (*Rapid Application Development*) dengan mengimplementasikan algoritma *Linear Congruential Generator*, dengan adanya aplikasi ini dapat menambah media pembelajaran bagi anak usia dini.
2. Dengan menerapkan metode *Linear Congruential Generator* soal yang keluar akan berbeda.
3. Aplikasi ini dapat menarik minat anak usia dini untuk belajar matematika.

#### 5.2 Saran

Dari hasil dan analisa perancangan sistem ini penulis menyadari bahwa masih banyak kekurangan dan kesalahan dalam perancangan sistem ini, dan berikut adalah beberapa saran yang diberikan untuk meningkatkan kemampuan dan fungsi dari sistem pada pengembangan yang lebih lanjut:

- a. Aplikasi ini hanya berjalan pada *smartphone* berbasis Android, diharapkan pada pengembangan selanjutnya aplikasi ini dapat berjalan pada *smartphone* lainnya seperti iOS ataupun sistem operasi yang lainnya.
- b. Diharapkan aplikasi ini dapat dikembangkan lebih lanjut dan penambahan level permainan dengan tingkat kesulitan yang sesuai.

## DAFTAR PUSTAKA

- Abidin, Z., Mustain, & Wahyudi, M. H. (2017). Game Tradisional Go Egrang Berbasis Android. *J-TIIES*, Vol. 1(1), 435-439.
- Adelia, & Setiawan, J. (2011). Implementasi Customer Relationship Management (CRM) pada Sistem Reservasi Hotel berbasis Website dan Dekstop. *Jurnal Sistem Informasi*, Vol. 6(2).
- Aswati, S., Ramadhan, M. S., Firmansyah, A. U., & Anwar, K. (2017). Studi Analisis Model Rapid Application Development Dalam Pengembangan Sistem Informasi. *Jurnal Matrik*, Vol. 16(2), 20-26.
- Busran, & Fitriyah. (2015). Perancangan Permainan (Game) Edukasi Belajar Membaca Pada Anak Prasekolah Berbasis Smartphone Android. *Jurnal TEKNOIF*, 62.
- Caesar, R. (2015). Kajian Pustaka Perkembangan Genre Games Dari Masa Ke Masa. *Journal of Animation and Games Studies*, 114-131.
- Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi Phonegap. *Jurnal String*, Vol. 3(2), 207.
- Fithri, D. L., & Setiawan, D. A. (2017). Analisa Dan Perancangan Game Edukasi Sebagai Motivasi Belajar Untuk Anak Usia Dini. *Jurnal SIMETRIS*, Vol. 8(1), 225-227.
- Handriyantini, E. (2009, June 24). Permainan Edukatif (Educational Games) Berbasis Komputer untuk Siswa Sekolah Dasar. Bandung, Jawa Barat, Indonesia.
- Hangga, A., & Prabowo, H. E. (2016). Modifikasi Linear Congruential Generator Untuk Sistem Pengacakan Soal Pada Computer Based Test (CBT). *Jurnal Teknik Elektro*, Vol. 8(2), 47-49.
- Hariyanto, D., Sastra, R., & Putri, F. E. (2021). Implementasi Metode Rapid Application Development Pada Sistem Informasi Perpustakaan. *Jurnal JUPITER*, Vol. 13(1), 110-112.
- Jogiyanto. (2005). *Analisis & Desain: Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta: Andi Offset.
- Julianus, E. (2019). Implementasi Game Online Multiplayer Pada Game Quick Quiz Menggunakan Metode Pengacakan Linear Congruential Generator (LCG). *Unknown*.
- Kosidin, & Farizah, R. N. (2016). Pemodelan Aplikasi Mobile Reminder Berbasis Android. *Seminar Nasional Teknologi Informasi dan Komunikasi*, 271-275.

- Kurniawan, H. (2017). Media Pembelajaran Mobile Learning Menggunakan Android. *Explore - Jurnal Sistem Informasi dan Telematika*, Vol. 8(No. 1), 46-49.
- Kusniyati, H., & Sitanggang, N. S. (2016). Aplikasi Edukasi Budaya Toba Samosir Berbasis Android. *Jurnal Teknik Informatika*, Vol. 9(1), 9-15.
- Ladjamudin, A.-B. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Madcoms. (2018). *Memfaatkan Aplikasi Pendukung Android Pada Sistem Operasi Windows*. Yogyakarta: Andi offset.
- Naproni. (2017). Perbandingan Algoritma Pembangkit Bilangan Linear Congruential Generator (LCG) dengan Multiplicative Random Number Generator (MNRG) dalam Probabilitas Kemunculan Bilangan yang Sama. *Seminar Nasional Teknologi Informatika*, 1-4.
- Nuqisari, R., & Sudarmilah, E. (2019). Pembuatan Game Edukasi Tata Surya dengan Construct 2 Berbasis Android. *Jurnal Teknik Elektro*, 86.
- Pane, B., Najoran, X., & Paturusi, S. (2017). Rancang Bangun Aplikasi Game Edukasi Ragam Budaya Indonesia. *E-Journal Teknik Informatika*, Vol. 12(1), 1-3.
- Pramuditya, S. A., Noto, M. S., & Purwono, H. (2018). Desain Game Edukasi Berbasis Android pada Materi Logika Matematika. *JNPM (Jurnal Nasional Pendidikan Matematika)*, Vol. 2(2), 165.
- Pramuditya, S. A., Noto, M. S., & Syaefullah, D. (2017). Game Edukasi RPG Matematika. *EduMa*, Vol. 6(1), 77.
- Ramadhani, R., & Siregar, R. F. (2022). Penerapan Model Pembelajaran Kooperatif Tipe STAD Menggunakan LAS Untuk Meningkatkan Hasil Belajar Matematika Siswa. *GENTA MULIA: Jurnal Ilmiah Pendidikan*, 339.
- Satyaputra, A., & Aritonang, E. M. (2014). *Beginning Android Programming with Adt Bundle*. Jakarta: Elex Media Komputindo.
- Syafari, & Sauda, S. (2020). Simulasi UN SMP Berbasis Web Menggunakan Metode Linear Congruential Generator (LCG). *Bina Darma Conference on Computer Science*, 259-263.
- Vitianingsih, A. V. (2016). Game Edukasi Sebagai Media Pembelajaran Pendidikan Anak Usia Dini. *Jurnal INFORM*, Vol. 1(No. 1).
- Wahyudi, R., & Pasaribu, H. H. (2015). Perancangan Aplikasi Quiz Menggunakan Metode Pengacakan Linear Congruential Generator (LCG) Berbasis Android. *Riau Journal Of Computer Science*, 17-22.

- Yunus, M., Astuti, I. F., & Khairina, D. M. (2015). Game Edukasi Matematika Untuk Sekolah Dasar. *Jurnal Informatika Mulawarman*, 59.
- Yustin, J. A., Sujaini, H., & Irwansyah, M. A. (2016). Rancang Bangun Aplikasi Game Edukasi Pembelajaran Matematika Menggunakan Construct 2. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, Vol. 1(1), 1.