**PROPOSAL**

**GAME EDUKASI BERBASIS ANDROID UNTUK ANAK USIA DINI**

## MENGGUNAKAN *LINEAR CONGRUENT METHOD* (LCM)

Diajukan Untuk Memenuhi

Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



**PRATIWI NUR AISYAH**

**E1E1 16 023**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALU OLEO**

**KENDARI**

**2020**

# DAFTAR ISI

**HALAMAN JUDUL............................................................................................i INTISARI.............................................................................................................ii ABSTRACT......................................................................................................... iii KATA PENGANTAR......................................................................................... iv DAFTAR ISI........................................................................................................v DAFTAR GAMBAR........................................................................................... vi DAFTAR TABEL................................................................................................vii**

## BAB I PENDAHULUAN.................................................................................... 1

1.1.Latar Belakang.....................................................................................1

1.2.Rumusan Masalah............................................................................... 3

1.3.Batasan Masalah..................................................................................3

1.4.Tujuan Penelitian.................................................................................3

1.5.Manfaat Penelitian...............................................................................4

1.6.Sistematika Penulisan..........................................................................4

1.7.Tinjauan Pustaka................................................................................. 4

## BAB II LANDASAN TEORI..............................................................................6

1. 1.Game ...................................................................................................6

2.1.1....Jenis Game...............................................................................8

2.1.2....*Game* Edukasi..........................................................................9

1. 2.Anak Usia Dini.................................................................................... 11

2.2.1....Pengertian Anak Usia Dini......................................................11

2.2.2....Karakteristik Anak Usia Dini.................................................. 11

2.2.3....Perkembangan Anak Usia Dini............................................... 12

2.2.4....Program Pendidikan Anak Usia Dini...................................... 14

2.3.Pembangkit Bilangan Acak................................................................. 19

2.4.*Linear Congruent Method*................................................................... 20

2.5.Uji Keacakan....................................................................................... 23

2.6.*Android*................................................................................................ 27

1. 7.*Unified Modelling Languange*.............................................................27

2.7.1....*Use Case Diagram*.................................................................. 28

2.7.2....*Activity Diagram*..................................................................... 30

2.7.3....*Sequence Diagram*...................................................................31

2.7.4....*Class Diagram*.........................................................................32

1. 8.Bahasa Pemrograman.......................................................................... 27

2.8.1....*Typescript*................................................................................ 34

2.8.2....*Angular CLI*.............................................................................35

1. 9.*Database*..............................................................................................35
   1. *SQlite*................................................................................................ 36
   2. *Android SDK* ................................................................................... 36
   3. Pengujian Blackbox..........................................................................37

## BAB III METODOLOGI PENELITIAN..........................................................39

3.1.Metode Pengumpulan Data................................................................. 39

3.2.Metode Pengembangan Aplikasi.........................................................39

1. 3.Waktu dan Tempat Penelitian............................................................. 41

3.3.1....Waktu Penelitian.......................................................................... 41

3.3.2....Tempat Penelitian.........................................................................41

3.4.Analisis Sistem.................................................................................... 42

3.4.1....Analisis Kebutuhan Fungsional..............................................42

3.4.1.1 Analisis Kebutuhan Input................................................ 42

3.4.1.2 Analisis Kebutuhan Proses.............................................. 42

3.4.1.3 Analisis Kebutuhan Output..............................................43

3.4.2... Analisis Kebutuhan Non-Fungsional......................................42

3.4.2.1 Analisis Kebutuhan Perangkat Keras.............................. 43

* + - 1. Analisis Kebutuhan Perangkat Lunak............................. 44

3.5 .*Unified Model Languange*.................................................................. 42

3.5.1... Perancangan *Use Case Diagram*............................................ 44

3.5.2....Perancangan *Activity Diagram*............................................... 46

3.5.3....Perancangan *Sequence Diagram*............................................ 47

3.5.4....Perancangan *Class Diagram*.................................................. 49

3.6.Metode Pengujian................................................................................ 49

3.7.Perancangan Algoritma *Linear Congruent Method*........................... 50

3.7.1....Kasus 1................................................................................... 52

3.7.2....Kasus 2................................................................................... 53

3.7.3....Kasus 3................................................................................... 54

3.8.Perancangan Sistem.............................................................................56

## DAFTAR PUSTAKA

**DAFTAR GAMBAR**

Gambar 3.1 *Use Case Diagram*................................................................... 45

Gambar 3.2 *Activity Diagram Game* Edukasi..............................................46

Gambar 3.3 *Activity Diagram Linear Congruent Method*........................... 46

Gambar 3.4 *Activity Diagram Game* Edukasi..............................................47

Gambar 3.5 *Sequence Diagram* Tentang..................................................... 47

Gambar 3.6 *Sequence Game Diagram* Edukasi...........................................48

Gambar 3.7 *Sequence Diagram* Keluar....................................................... 48

Gambar 3.8 *Class Diagram Game* Edukasi................................................. 49

Gambar 3.9 *Flowchart* Algoritma *Linear Congruent Method*.....................50

Gambar 3.10 Tampilan *Dashboard*............................................................. 57

Gambar 3.11 Tampilan Mulai Game........................................................... 58

Gambar 3.12 Tampilan Kuis Huruf ............................................................ 58

Gambar 3.13 Tampilan Skor Tiap Kuis....................................................... 59

**DAFTAR TABEL**

Tabel 2.1 Optimalisasi Perkembangan Kognitif pada Anak Usia Dini....... 17

Tabel 2.2 Tabel A.5 ( شم) dengan α = 0,05................................................. 25

Tabel 2.3 Tabel A.6 ( شم) dengan α = 0,05................................................. 26

Tabel 2.4 Hasil Pengkategorian Data...........................................................27

Tabel 2.5 Simbol *Use Case Diagram*.......................................................... 29

Tabel 2.6 Simbol *Activity Diagram*............................................................. 31

Tabel 2.7 Simbol S*equence Diagram*.......................................................... 32

Tabel 2.8 Simbol *Class Diagram*.................................................................32

Tabel 3.1 *Gantt Chart* Waktu Penelitian..................................................... 62

Tabel 3.1 Spesifikasi Perangkat Keras.........................................................43

Tabel 3.2 Spesifikasi Perangkat Lunak........................................................44

Tabel 3.3 Definisi *Use Case Diagram*......................................................... 45

Tabel 3.4 Rancangan Pengujian Aplikasi 5................................................. 49

Tabel 3.5 Soal Quiz......................................................................................58

**BAB I**

**PENDAHULUAN**

## 1.1 Latar Belakang

Seiring dengan perkembangan teknologi informasi, sebuah media pembelajaran kini mulai di kembangkan dengan memadukan dunia pendidikan dan dunia hiburan. Hal ini dilakukan karena peserta didik akan lebih tertarik untuk mempelajari dunia pendidikan jika di dalam proses pembelajaran dapat membuat mereka senang untuk belajar tanpa adanya perasaan terpaksa. Menurut Desmita (dalam Novan Ardy, 2002:114) Perkembangan pada anak usia dini terkait dengan perubahan psikis pada diri mereka, yang mencangkup aspek kognitif, bahasa, sosial, dan emosi, serta agama dan moral. Perkembangan.kognitif pada anak usia dini tetkait dengan pengetahuannya, yaitu semua proses psikologis yang berhubungan dengan bagaimana anak mempelajari dan memikirkan lingkungannya.

*Game* merupakan penarik perhatian yang telah terbukti. *Game* adalah lingkungan pelatihan yang baik bagi dunia nyata dalam organisasi yang menuntut pemecahan masalah secara kolaborasi. (John C Beck & Mithcell Wade, 2004). *Game* edukasi unggul dalam beberapa aspek jika di bandingkan dengan metode pembelajaran konvesional. 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 pembelajaran konvensional (Donal Clark : 2006).

Berdasarkan hasil wawancara dengan Kepala Sekolah TK Pembina 1 Kendari, diperoleh informasi bahwa perkembangan teknologi yang semakin pesat maka kita juga harus mengenalkan teknologi kepada anak. Dengan adanya *game* edukasi kita bisa memperkenalkannya dengan cara menanamkan 6 aspek perkembangan anak dalam *game* edukasi tersebut. 6 aspek yaitu : aspek nilai agama dan moral, aspek fisik dan motorik, aspek kognitif, aspek sosial emosional, aspek bahasa, dan aspek seni

Perkembangan teknologi *game*, sangat perlu dikembangkannya *game* edukasi untuk anak usia dini. Anak usia dini adalah anak yang berada pada usia 0 sampai 8 tahun yang sedang dalam tahap pertumbuhan dan perkembangan, baik fisik maupun mental (Augusta, 2012). Karakteristik anak usia dini adalah suka berimajinasi dan berfantasi, hal ini penting bagi pengembangan kreativitas dan kemampuan bahasa.

Berdasarkan permasalahan di atas perlu dipandang untuk dikembangankan *game* edukasi bagi anak usia dini. Untuk ini peneliti membuat *game* edukasi yang meliputi pengenalan huruf, angka, gambar, suara dan warna untuk mengembangkan keterampilan kognitif dan motorik anak, melatih kesabaran, mengembangkan kordinasi mata dan tangan, dan membantu meningkatkan kemampuan anak dalam memecahkan masalah. *Game* edukasi yang peneliti buat terdiri dari pengenalan huruf berupa menyusun kata dari huruf-huruf yang disediakan, pengenalan angka dengan model kuis penyusunan angka, pengenalan gambar melalui suara dan pengenalan suara berdasarkan gambar yang ditampilkan, serta pengenalan warna melalui gambar 2 dimensi.

Dalam sebuah permainan edukasi, soal yang ditampilkan dalam setiap sesi harus dibuat teracak untuk membuat *game* lebih menarik dan menantang dan tidak membosankan bagi anak-anak dalam bermain *game.* Metode pengacakan saat ini sudah banyak digunakan dalam sistem berbasis teknologi. Salah satu contoh metode yang dapat membangkitkan bilangan acak adalah *Linear Congruent Method.* Metode ini dapat membangkitan kumpulan bilangan acak dengan proses yang relatif cepat dibandingkan dengan metode serupa (Andresta, 2017). Untuk itu digunakan *Linear Congruent Method (LCM)* untuk membangkitkan bilangan acak sehingga soal pada *game* edukasi bagi anak usia dini pada setiap sesinya tidaklah sama dan pilihan jawaban yang ada akan ditampilkan teracak. *Game* juga akan menampilkan hasil jawaban dari pemain, sehingga pemain (anak-anak) dapat belajar mengenai jawaban mereka sebelumnya dan dapat memperbaiki kesalahan di *game* berikutnya.

Berdasarkan uraian diatas, penulis tertarik untuk melakukan penelitian dengan judul **“*Game* Edukasi Berbasis Android untuk Anak Usia Dini Menggunakan *Linear Congruent Method (LCM)*”.** Penelitian ini bermanfaat untuk membuat *game* yang mengajarkan dan mengembangkan keterampilan kognitif dan motorik anak usia dini.

## 1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut : 1. Bagaimana membangun aplikasi *game* edukasi mengenal huruf, angka, gambar, suara dan warna berbasis android untuk anak usia dini.

2. Bagaimana mengimplementasikan algoritma *Linear Congruent Method (LCM)* untuk pengacakan soal pada *game* edukasi.

## 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut :

1. *Game* edukasi ini terdiri dari pengenalan huruf, penyusunan huruf menjadi kata, pengenalan angka, penjumlahan angka, pengenalan gambar hewan, pengenalan suara hewan, dan pengenalan warna.
2. Dalam pembuatannya menggunakan bahasa pemrograman *TypeScript*, *framework* yang digunakan *ionic* dan penyimpanan data menggunakan *SQLite*.
3. Algoritma yang digunakan adalah *Linear Congruent Method (LCM).*
4. Aplikasi berjalan secara *offline.*
5. Umur anak yang akan menggunakan game adalah 3 - 6 Tahun
6. Fokus penelitian adalah implementasi algoritma pengacakan pada soal quiz.
7. Monitoring peningkatan kognitif anak tidak dapat dilakukan menggunakan system.
8. Aplikasi tidak berjalan untuk *multiuser*.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Bagaimana membangun aplikasi *game* edukasi mengenal huruf, angka, gambar, suara dan warna berbasis android untuk anak usia dini.
2. Bagaimana mengimplementasikan algoritma *Linear Congruent Method (LCM)* untuk pengacakan soal pada *game* edukasi.

## 1.5 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut :

1. Meningkatkan motivasi dan hasil belajar anak pada materi dasar melalui pembelajaran yang aktif, kreatif, efektif dan menyenangkan.
2. Permainan berbasis android ini dapat digunakan sebagai salah satu media pengajaran.
3. Membuat *game* edukasi yang implementasinya untuk anak dalam belajar, dan mengimplementasikan algoritma *Linear Congruent Method (LCM)* untuk pengacakan soal pada *game* edukasi.

**1.6 Sistematika Penulisan**

Sistematika penulisan proposal Tugas Akhir ini adalah sebagai berikut:

## BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

## BAB II LANDASAN TEORI

Bab ini memuat pengertian-pengertian dan teori-teori yang menjadi acuan dalam pembuatan analisa dan pemecahan dari permasalahan yang dibahas meliputi *Game*, Anak Usia Dini, Pembangkit Bilangan Acak, *Linear Congruent Method (LCM)*, Android, *Unified Modeling Language* (UML), Bahasa Pemrograman, *Database, SQLite, Android SDK,* Pengujian Mobile.

## BAB III METODOLOGI PENELITIAN

Bab ini berisi metode penelitian yang digunakan. Metode pengembangan perangkat lunak yang digunakan adalah *Rational Unified Proses* (RUP), sedangkan penyusunan laporan menggunakan studi literatur.

## BAB IV ANALISIS DAN PERANCANGAN

Bab ini akan membahas tentang analisis dan perancangan dari aplikasi yang akan dibuat, dengan menggunakan desain Unified Modelling Language (UML) serta desain user interface.

## BAB V IMPLEMENTASI DAN PEMBAHASAN

Dalam bab ini akan dikaji mengenai implementasi hasil perancangan aplikasi yang dibuat serta melakukan pengujian terhadap sistem.

## BAB VI PENUTUP

Bab ini berisi kesimpulan dari program yang telah dibuat serta saran yang diperlukan untuk pengembangan program berikutnya.

### 1.7 Tinjauan Pustaka

Dalam suatu penetian diperlukan dukungan hasil-hasil penelitian yang telah ada sebelumnya yang berkiatan dengan penelitian tersebut. Berdasarkan penelitian Dian Wahyu Putra (2016), melakukan penelitian dengan judul “*Game* Edukasi Berbasis Android sebagai Media Pembelajaran untuk Anak Usia Dini”. Penelitiannya mengenai membuat *game* edukasi untuk anak usia dini mengenal binatang, mewarnai, menyanyi serta alfabet menggunakan aplikasi *AppInventor.*

Penelitian oleh Elys Candra Yani (2016) mengenai Rancang Bangun

Aplikasi *Edugame Slider Puzzle* Keanekaragaman Budaya Bengkulu dengan Menggunakan *Linear Congruent Method (LCM)* di Universitas Bengkulu. Penelitian ini mengembangkan ketrampilan kognitif dan motorik anak dengan tingkat kesulitan yaitu mudah, sedang dan sulit yang memiliki jumlah soal yang berbeda. Uji prestasi kemampuan pengguna menunjukkan bahwa aplikasi ini memiliki pengaruh dalam membantu meningkatkan kemampuan kognitif dan pengenalan budaya.

Pada tahun 2017, Muhammad Ganda Arizqia dan Anang Aris Widodo meneliti tentang Rancang Bangun Aplikasi dengan *Linear Congruent Method* Sebagai pengacakan soal. Cara yang digunakan untuk menyusun soal adalah dengan mengumpulkan soal-soal yang telah dibuat kemudian diinputkan oleh *user* pada aplikasi dan siap diacak melalui aplikasi. Untuk menerapkan *linear congruent method* (LCM) untuk pengacakan soal-soal, ada hal-hal yang harus diperhatikan. Penggunaan konstanta *a, c*, dan *m* sangat menentukan pengacakan yang terjadi sehingga dengan kombinasi konstanta yang tepat maka akan dihasilkan pengacakan soal yang benar-benar teracak.

**BAB II**

**LANDASAN TEORI**

## 2.1 *Game*

Berdasarkan kamus, *game* berasal dari kata bahasa inggris yang berarti permainan. Banyak teori yang mengungkapkan tentang pengertian *game*. *Game* merupakan suatu sistem atau program dimana satu atau lebih pemain mengambil keputusan melalui kendali pada obyek didalam *game* untuk suatu tujuan tertentu. Teori lain mengatakan bahwa *game* merupakan salah satu media hiburan yang menjadi pilihan masyarakat untuk menghilangkan kejenuhan atau untuk mengisi waktu luang. Selain sebagai media hiburan, *game* juga dapat meningkatkan perkembangan otak seseorang, contohnya adalah permainan catur yang dapat meningkatkan konsentrasi otak (Yandi, 2014).

Berdasarkan pengertian tersebut dapat dikatakan bahwa *game* atau permainan merupakan suatu media yang digunakan untuk tujuan hiburan dimana pemainnya melakukan suatu kegiatan guna mencapai tujuan tertentu yang biasanya dapat dilakukan secara mandiri atau *single player* maupun lebih dari satu pemain atau *multiplayer*. Proses pembuatan *game* biasanya pembuat *game* memiliki suatu tujuan tertentu yang disesuaikan dengan kebutuhan dan keinginan pembuat *game*. Tujuan pembuatan *game* antara lain (Yandi, 2014) : a. Sebagai hiburan (*Entertainment*)

1. Melatih ketangkasan (*Expand Skill*)
2. Mendidik (*Education*)
3. Menyampaikan pesan (*Embed Messages*)

### 2.1.1 Jenis *Game*

Jenis *game* umumnya disebut juga dengan istilah *genre game* yang dapat diklasifikasikan sebagai berikut (Sagala, 2009) :

6

a. *Maze Game*

Jenis game ini adalah jenis *game* yang paling awal muncul. Secara sederhana kita hanya mengitari *maze* (lorong-lorong yang berhubungan) dan memakan beberapa item untuk menambah tenaga atau kekebalan. Kita juga

tentunya memiliki musuh yang mengejar kita. Ketika kita mendapatkan kekebalan kita bisa berbalik mengejar mereka mereka. Mode permainan inilaiS yang menjadi dasar dari permainan 3D sekarang. Contoh maze *game* ialah *Digger, Pacman, Doom, Ultimate Doom, Quake.* b. *Board Game*

Jenis *game* ini sama dengan *game* board tradisional, seperti *Monopoly*. Sampai saat ini tidak ada variasi yang memunculkan gameplay ataupun perubahan desain dari versi tradisional ke versi elektronik. Versi elektronik benar-benar hanya memindahkan versi tradisional ke layar komputer. Umumnya *game* ini lebih menekankan kepada kemampuan komputer menjadi lawan tanding dari pemain. Contoh *game* jenis ini ialah *Chess, Monopoly, Scrabble*. c. *Card Game*

Hampir sama dengan *board game*, genre ini tidak memberikan perubahan berarti dari game tradisional yang sejenis. Contohnya, game *Solitaire dan Hearts,* versi asli dan elektroniknya nyaris tidak ada bedanya. Variasinya adalah kemampuan *multiplayer* dan tampilan yang lebih bervarisi dari versi tradisional. *Game* ini termasuk game yang muncul pada awal *game* komputer seperti genre maze dan board *game*. Contohnya, *Hearts, Spider, Balckjack, Poker, Solitaire*. d. *Quiz Game*

Jenis ini juga agak jarang di Indonesia. Salah satu yang umum dikenal adalah *game* kuis *Who Wants To Be Milionare* sebuah *game* dengan nama yang sama dari acara kuis televisi. *Game* ini sederhana dalam cara bermain. Kita hanya perlu memilih jawaban yang benar dari beberapa pilihan jawaban yang benar dari beberapa pilihan jawaban. Biasanya pertanyaan yang diberikan memang memiliki topik tertentu. Contoh lainnya ialah, *Clasrooom Milionare, Deal or No Deal*. e. *Puzzle Game*

*Game* ini memberikan tantangan kepada pemainnya dengan cara menjatuhkan sesuatu dari sisi sebelah atas ke bawah. Pemain harus menyusunnya sedemikian rupa dan tidak ada yang tersisa ketika susunan diatas sudah akan dibuat. Susunan ini dilakukan secepat dan sebaik mungkin. Semakin lama akan semakin cepat dan semakin banyak objek yang jatuh. Dalam perkembangan, jenis ini membebaskan cara bermainnya. Misalnya dengan *user* bebas meletakkan objek ke suatu tempat dengan tujuan tertentu. Contoh *game* ini ialah *Magic Inlay, Adventure Inlay, Tetris, Chip Challenge.* f. *Sport Game*

*Game* ini sama dengan *game* endutaiment. Genre ini hanya berdasarkan jenisnya, bukan berdasarkan teknologi atau spesifikasi teknis apapun. Selama *game* itu mengetengahkan genre olahraga maka disebut *genre sport* tidak peduli apakah *game* ini menggunakan gaya arcade 2D atau 3D. Contoh : *Winning Eleven, Championship Manager, FIFA, NBA, Virtual Tennis.* g. *Racing Game*

*Game* balapan, *game* ini memberikan permainan lomba kecepatan dari kendaraan yang dimainkan oleh pemain. Terkadang didalam arena, terkadang diluar arena balap. Contoh *game* ini yaitu, *Need For Speed, NFS Underground, Driver, Taxi.*

h. *Simulation Game*

Disini kita sebagai pemain membangun secara simulasi sebuah kota, negara atau koloni. Kita mengatur berbagai sumber daya dan menentukan berbagai keputusan yang kita inginkan dalam proses pembangunan yang sedang terjadi. Disamping itu pula terdapat *sub-genre* dari *simulation game* seperti *flight simulator* dan *technical simulator.* i. *Turn Based Strategy Game*

*Game* ini memerlukan strategi dari pemain untuk memenangkan permainan. Pemain melakukan gerakan setelah pemain lain melakukannya, jadi saling bergantian. Bisa dibilang mirip dengan catur, tetapi dengan variasi gerakan dan efek yang jauh lebih banyak. Contoh: *Empire, Civilization, Heroes of Might and Magic.*

1. *Real Time Strategy game*

Jika pada *Turn Based Strategy game* kita perlu menunggu pemain lain menyerang, pada *Real Time Strategy game* ini kita tidak perlu menunggu. Pemain yang tercepatlah yang besar kemungkinannya untuk menang. Pada jenis *game* ini kita harus melakukan beberapa gerakan sesuai dengan strategi kita.

Contoh *game* jenis ini ialah *Warcraft, Starcraft, Commandos, Command and Conquer.*

1. *Role Playing Game*

*Genre game* ini kita akan berperan menjadi sebuah karakter. Kita akan menjalankan peran kita ini dengan berbagai atribut, seperti kesehatan, intelegensi, kekuatan, dan keahlian. Salah satu *game* yang terkenal dengan RPG pada masa awal adalah Ultima. Kini *genre* ini berkembang menjadi beberapa jenis variasi RPG seperti action *Role Playing Game.*

Berdasarkan jenis *game* tersebut, jenis *game simulation game, sport game, role playing game, real time strategy dan racing* yang paling popular dari kalangan anak-anak hingga remaja di industri *game* baik *mobile* maupun komputer. Untuk *game* kategori *puzzle dan quiz* lebih ditujukan kepada anakanak sebagai media pengenalan objek dan edukasi (Sagala, 2009).

### 2.1.2 *Game* Edukasi

Penerapan *game* edukasi bermula dari perkembangan *video game* yang sangat pesat dan menjadikannya sebagai media yang efektif, interaktif dan banyak dikembangkan di perusahaan industri. Melihat kepopuleran game tersebut, para pendidik berpikir bahwa komponen rancangan *game* dapat digunakan sebagai pengantar materi pembelajaran serta dapat diterapkan pada kurikulum pendidikan. *Game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan. Kriteria yang harus dimiliki dari *game* edukasi sebagai berikut (Hurd, 2009):

1. Nilai Keseluruhan

Nilai keseluruhan berhubungan dengan semua hal yang terdapat dalam *game* edukasi misalnya cara bermain, *game* edukasi dapat dimainkan kembali dan biaya pembuatan harus diperhatikan dengan baik.

1. Kegunaan (*Usability*)

Kegunaan berhubungan dengan seberapa baik *game* edukasi dapat memberikan pengetahuan bagi penggunanya.

1. Keakuratan (*Accuracy*)

Keakuratan berhubungan dengan kesesuaian konten yang terdapat dalam *game* edukasi pada proses pembelajaran, sehingga dapat menambah ilmu pengetahuan penggunanya.

1. Kelayakan

Kelayakan berhubungan dengan bagaimana konten dan desain *game* edukasi disesuaikan dengan sasaran pengguna. Suatu *game* edukasi dikatakan layak jika tujuan dari *game* edukasi yaitu untuk memberikan keahlian dan pengetahuan khusus bagi pengguna tercapai.

1. Hubungan (*Relationship*)

Kriteria ini berhubungan dengan bagaiman suatu konten yang ada dalam *game* edukasi. Hal ini bertujuan agar *game* edukasi yang dibuat dapat dimainkan dengan baik sesuai dengan umur dan karakteristik pengguna.

1. Tujuan

Kriteria ini berhubungan dengan apa yang didapat dari memainkan game edukasi. *Game* edukasi harus dapat memberikan manfaat bagi penggunanya sehingga konten yang ada dalam *game* edukasi harus jelas, layak dan bersifat objektif.

1. Umpan Balik

*Game* edukasi harus memberikan umpan balik yang bersifat positif misalnya pemberian efek suara, indikasi benar atau salah, keterangan setelah menyelesaikan *game* dan sebagainya.

1. Kesenangan

Kriteria ini berhubungan dengan bagaimana pengguna dapat menikmati pemainan yang ada di dalam *game* edukasi. *Game* edukasi harus dapat memberikan kesenangan dan ketertarikan bagi pengguna. *Game* edukasi diharapkan dapat meningkatkan ilmu pengetahuan dan keahlian penggunanya.

## 2.2 Anak Usia Dini

### 2.2.1 Pengertian Anak Usia Dini

Anak adalah manusia kecil yang memiliki potensi yang masih harus dikembangkan. Anak memiliki karakteristik tertentu yang khas dan tidak sama dengan orang dewasa serta akan berkembang menjadi manusia dewasa seutuhnya. Menurut Undang-undang Sisdiknas tahun 2003 “anak usia dini adalah anak yang berada pada rentang usia 0-6 tahun”. Anak memiliki karakteristik tertentu yang khas dan tidak sama dengan orang dewasa. Anak selalu aktif, dinamis, antusias dan ingin tahu terhadap apa yang dilihat dan didenganya, seolah-olah tidak pernah berhenti belajar (Kementerian Pendidikan dan Kebudayaan, 2003). Mansur mengungkapkan bahwa “anak usia dini adalah kelompok anak yang berada dalam proses pertumbuhan dan perkembangan yang khusus sesuai dengan tingkat pertumbuhan dan perkembangan” (Mansur, 2005). Sejalan dengan hal tersebut, Hartati mengungkapkan bahwa “anak usia dini adalah seorang manusia atau individu yang memiliki pola perkembangan dan kebutuhan tertentu yang berbeda dengan orang dewasa” (Hartati, 2005). Berdasarkan ungkapan di atas maka dapat disimpulkan bahwa, anak usia dini merupakan individu berusia 0-6 tahun yang memiliki ciri khas unik dan sedang dalam tahap pertumbuhan serta perkembangan, baik fisik maupun mental. Pendidikan anak usia dini akan membantu proses pertumbuhan dan perkembangan anak menjadi manusia yang lebih baik menuju kematangan.

### 2.2.2 Karakteristik Anak Usia Dini

Anak usia dini memiliki karakteristik yang berbeda dengan orang dewasa, beberapa karakteristik anak usia dini yaitu (Hartati, 2005): a. Anak bersifat egosentris

1. Anak memiliki rasa ingin tahu yang besar
2. Anak adalah mahluk sosial
3. Anak bersifat unik
4. Anak umumnya kaya dengan fantasi
5. Anak memiliki daya konsentrasi yang pendek
6. Anak merupakan masa belajar yang paling potensial

Sejalan dengan ungkapan tersebut menjelaskan bahwa anak usia dini memiliki karakteristik yaitu : a. Bersifat egosentris naif

1. Mempunyai relasi sosial dengan benda-benda dan manusia yang sifatnya sederhana dan primitif
2. Ada kesatuan jasmani dan rohani yang hampir tidak terpisahkan sebagai satu totalitas
3. Sikap hidup yang fungsionis

Berdasarkan beberapa ungakapan diatas dapat disimpulkan bahwa anak memiliki karakteristik yang jauh berbeda dengan orang dewasa. Anak adalah sosok individu yang sangat aktif, dinamis antusias dan selalu ingin tahu terhadap apa yang dilihat dan didengarnya.

### 2.2.3 Perkembangan Anak Usia Dini

Perkembangan anak usia dini memiliki beberapa aspek yaitu aspek fisikmotorik, kognitif, bahasa, seni dan sosial-emosional. Aspek- aspek tersebut tidak dapat berkembang sendiri-sendiri, melainkan aspek-aspek tersebut saling berkaitan.apabila salah satu aspek tidak dapat berkembang dengan baik maka aspek-aspek yang lainnya juga terhambat perkembangannya. Dibawah ini adalah aspek-aspek tersebut (Hartati, 2005): a. Perkembangan Fisik-Motorik

Aspek perkembangan fisik-motorik mencakup pertumbuhan fisik pada setiap anak tidak selalu sama, ada beberapa anak yang mengalami pertumbuhan secara cepat, dan ada pula yang mengalami kelambatan. Pada usia yang sama juga kadang kita temukan satu anak memiliki badan yang tinggi dan anak lainnya lebih pendek. Pada masa usia dini, pertumbuhan tinggi dan berat badan relatif seimbang, tetapi secara bertahap tubuh anak akan mengalami perubahan.

Apabila dimasa bayi anak memiliki penampilan yang gemuk maka secara perlahan tubuhnya akan berubah menjadi lebih langsing, sedangkan kaki dan tangannya mulai memanjang. Perkembangan motorik anak juga sudah berkembang baik. Jika pada usia 1 tahun anak ada yang belum terampil berjalan, maka pada usia 2,5 tahun anak umumnya sudah dapat berlari, melompat, menendang bola, dan memanjat. Setiap gerakannya sudah selaras dengan kebutuhan atau minatnya. Pada masa ini ditandai dengan kelebihan gerak atau aktivitas. Anak cenderung menunjukkan gerakan-gerakan motorik yang cukup gesit dan lincah. Oleh karena itu, usia ini merupakan masa ideal untuk belajar keterampilan yang berkaitan dengan motorik, seperti menulis, menggambar, melukis, berenang, main bola dan atletik. b. Perkembangan Kognitif

Perkembangan kognitif menyangkut perkembangan berfikir dan bagaimana kegiatan berfikir itu bekerja. Dalam kehidupannya, mungkin saja anak dihadapkan pada persoalan yang menuntut adanya pemecahan. Menyelesaikan suatu persoalan merupakan langkah yang lebih kompleks pada diri anak. Sebelum anak mampu menyelesaikan persoalan, anak perlu memiliki kemampuan untuk mencari cara penyelesainnya. Faktor kognitif mempunyai peranan yang penting bagi keberhasilan anak dalam belajar karena sebagian besar aktifitas dalam belajar selalu berhubungan dengan masalah mengingat dan berfikir. Perkembangan struktur kognitif berlangsung menurut urutan yang sama bagi semua anak. c. Perkembangan Bahasa

Bahasa merupakan sarana berkomunikasi dengan orang lain. Dalam pengertian ini tercakup semua cara untuk berkomunikasi, dimana pikiran dan perasaan dinyatakan dalam bentuk tulisan, lisan, isyarat atau gerak dengan menggunakan kata-kata, kalimat bunyi, lambang, gambar atau lukisan. Menurut Miller, bahasa adalah suatu urutan kata-kata, bahasa juga dapat digunakan untuk menyampaikan informasi mengenai tempat atau waktu yang berbeda. d. Perkembangan Sosial Emosional

Pada usia dini, emosi anak mulai matang. Anak mulai menyadari akibatakibat dari tampilan emosinya. Anak mulai memahami perasaan orang lain, misalnya bagaimana perasaan orang lain apabila disakiti maka anak belajar mengendalikan emosinya.

e. Perkembangan Seni

Melalui seni anak dapat mengembangkan beberapa aspek perkembangan lainnya seperti menyanyi sambil belajar huruf dan angka untuk membantu mengembangkan aspek perkembangan kognitif atau menggunting, menggambar dan menari untuk mengembangkan aspek perkembangan kognitif, fisik, dan motorik anak. Kemampuan anak usia dini untuk merasakan dan melakukan berbagai keterampilan atau kemampuan seninya dapat ditimbulkan dan dikembangkan sejak dini melalui pelatihan dan bimbingan yang terarah sambil disesuaikan dengan karakteristik belajar anak usia dini yaitu bermain.

### 2.2.4 Program Pendidikan Anak Usia Dini

Anak usia dini yang biasa disingkat dengan AUD adalah anak yang berusia 0 hingga 6 tahun sebelum memasuki masa sekolah dasar. Tahapan yang dilalui anak pada masa usia dini adalah masa bayi dari usia lahir sampai dengan 12 bulan, masa kanak-kanak dari usia 1 tahun hingga 3 tahun. Selanjutnya masa pra sekolah dari usia 3 tahun sampai dengan usia 6 tahun. Berbagai aspek pertumbuhan dan perkembangan yang meliputi aspek pertumbuhan fisik, perkembangan kognitif, emosi, sosial, bahasa serta perkembangan moral dan agama.

Bidang garapan pendidikan anak usia dini meliputi tiga jenis yang masingmasing dibagi menurut usia anak. Lembaga pendidikan yang dimaksud terdiri atas:

Pendidikan Keluarga yang dikenal dengan nama Taman Penitipan Anak (TPA), Kelompok Bermain (KB) atau *play group*, dan Taman Kanak-kanak (TK).

Pendidikan Keluarga untuk anak usia 0 sampai 2 tahun, merupakan lingkungan yang pertama dan utama bagi anak sebab pendidikan keluarga merupakan fondasi bagi anak untuk membangun struktur kepribadian selanjutnya.

Taman Pengasuhan Anak (TPA) untuk anak usia 3 bulan sampai dengan 3 tahun. TPA adalah lembaga kesejahteraan sosial yang memberikan layanan pengganti berupa asuhan, perawatan dan pendidikan bagi anak balita selama anak tersebut ditinggal bekerja oleh orang tuanya. Hal ini untuk menghidarkan anak dari kemungkinan terlantar pertumbuhan dan perkembangan jasmani, rohani dan sosialnya. Kelompok Bermain (KB) atau *play group* untuk anak usia 3 sampai 4 tahun, yang merupakan tempat bermain dan belajar bagi anak-anak sebelum memasuki TK yang bertujuan mengembangkan seluruh aspek fisik, mental, emosi dan sosial anak. Isi program merupakan penjabaran dari visi dan misi serta tujuan kelompok bermain.

Taman Kanak-kanak (TK) adalah program untuk anak usia 4 sampai 6 tahun, yang merupakan jenjang pendidikan setelah kelompok bermain/play group sebelum anak masuk sekolah dasar. Pada saat ini taman Kanak-kanak bukan jenjang pendidikan wajib, dan tidak termasuk dalam program wajib belajar pendidikan dasar. Meskipun demikian, keberadaannya telah memberikan sesuatu yang cukup berarti bagi penyiapan anak usia dini memasuki pendidikan dasar.

Bina Keluarga Balita (BKB) adalah suatu kegiatan yang bertujuan memberikan pengetahuan dan keterampilan kepada orang tua dan anggota keluarga lainnya mengenai bagaimana mendidik, mengasuh dan memantau pertumbuhan dan perkembangan anak. Program ini diperuntukkan bagi ibu-ibu yang memiliki anak usi dini dan termasuk dalam kategori keluarga berpenghasilan rendah). Tujuan BKB adalah agar orang tua memiliki konsep diri yang sehat, terjadi peningkatan pengetahuan dan keterampilan dalam mengasuh dan membina anak serta mampu menerapkan pola asuh yang tepat sejak dini (Mulyasa, 2012:

55).

Undang-undang Nomor 20 tahun 2003 tentang Sistem Pendidikan Nasional, PAUD memiliki bagian tersendiri, yaitu pada Bagian Ketujuh pasal 28 dinyatakan dalam 6 ayat yang meliputi: (1) Pendidikan anak usia dini diselenggarakan sebelum jenjang pendidikan dasar. (2) Pendidikan anak usia dini dapat diselenggarakan melalui jalur pendidikan formal, nonformal, dan/atau informal. (3) Pendidikan anak usia dini pada jalur pendidikan formal berbentuk Taman Kanak-Kanak (TK), Raudatul Athfal (RA), atau bentuk lain yang sederajat.

(4) Pendidikan anak usia dini pada jalur pendidikan nonformal berbentuk Kelompok Bermain (KB), Taman Penitipan Anak (TPA), atau bentuk lain yang sederajat. (5) Pendidikan anak usia dini pada jalur pendidikan informal berbentuk pendidikan keluarga atau pendidikan yang diselenggarakan oleh lingkungan. (6) Ketentuan mengenai pendidikan anak usia dini sebagaimana dimaksud pada ayat

(1), ayat (2), ayat (3), dan ayat (4) diatur lebih lanjut dengan Peraturan Pemerintah. Taman Kanak-kanak (TK) sebagaimana dinyatakan dalam Undang-undang Republik Indonesia Nomor 20 tahun 2003 pasal 28 ayat 3 merupakan pendidikan anak usia dini pada jalur pendidikan formal yang bertujuan membantu anak didik mengembangkan berbagai potensi baik psikis dan fisik yang meliputi moral dan nilai agama, sosial, emosional, kemandirian, kognitif, bahasa, fisik/motorik dan seni untuk siap memasuki sekolah dasar. Selanjutnya dalam Peraturan Pemerintah Nomor 27 tahun 1990 Bab I pasal 1 ayat (2) dinyatakan bahwa “Taman KanakKanak adalah satu bentuk pendidikan prasekolah yang menyediakan program pendidikan dini bagi anak usia dini bagi anak usia empat sampai memasuki pendidikan dasar”.

Menurut Petunjuk Teknis Penyelenggaraan Taman Kanak-kanak, pengertian Taman Kanak-Kanak yang selanjutnya disingkat TK adalah salah satu bentuk satuan PAUD pada jalur pendidikan formal yang menyelenggarakan program pendidikan bagi anak berusia 4 tahun sampai dengan 6 tahun dengan prioritas usia 5 dan 6 tahun (Kemendikbud, 2015: 3).

Perkembangan pada anak usia dini terkait dengan perubahan psikis pada diri mereka, yang mencangkup aspek kognitif, bahasa, social, dan emosi, serta agama dan moral. Perkembangan kognitif pada anak usia dini terkait dengan pengetahuannya, yaitu semua proses psikologis yang berhubungan dengan bagaimana anak mempelajari dan memikirkan lingkungannya (Rosda, 2009). Kata kognitif berasal dari *cognition* yang sinonimnya adalah *knowing* yang berarti mengetahui.

Ada beberapa teori yang menjelaskan bagaimana perkembangan kognitif pada anak usia dini. Pertama, teori tahap perkembangan kognitif menurut Jean Piaget mengungkapkan bahwa manusia dalam hidupnya pasti melalui empat tahap perkembangan kognitif, di mana masing-masing tahap terkait dengan usia dan terdiri dari cara berpikir yang khas atau berbeda. Keempat tahap tersebut antara lain tahap *sensorimotor*, pra operasional konkret, dan tahap operasional formal.

Anak usia dini berada pada tahap *sensorimotor* dan tahap pra operasional. Anak yang berusia 0 hingga 2 tahun berada pada tahap *sensorimotor*. Tahap ini ditandai dengan kemampuan anak dalam melakukan gerak refleks sederhana (lahir sampai 1 bulan), belajar melakukan gerakan secara berulang-ulang (1-4 bulan), mulai menirukan gerakan sederhana (4-8 bulan), melihat suatu benda dan terdorong untuk menggenggamnya (8-12 bulan), mulai memiliki rasa ingin tahu dan mulai muncul minat (12-18 bulan), dan mampu menggunakan simbol-simbol sederhana (18-24 bulan).

Kemudian pada tahap pra operasional (2-4 tahun) merupakan tahap awal pembentukan konsep secara stabil. Penalaran mental mulai muncul, egosentrisme mulai kuat dan kemudian lemah, serta keyakinan terhadap hal yang magis terbentuk.

Pemberian stimulasi dalam optimalisasi perkembangan kognitif pada anak usia dini memiliki kemampuan berikut ini :

**Tabel 2.1 Optimalisasi Perkembangan Kognitif pada Anak Usia Dini**

|  |  |  |
| --- | --- | --- |
| **USIA** | **KEMAMPUAN KOGNITIF** | |
| 0-3 Bulan | * Mampu membedakan apa yang diinginkan (ASI, susu botol, atau kempong/*pacifier*). * Berhenti menangis setelah digendong atau diberi susu. | dari |
| 3-6 Bulan | * Memperlihatkan dan memilih permainan yang diinginkan. * Mengelurkan kedua tangan untuk digendong. |  |
| 6-9 Bulan | * Mengamati benda-benda yang bergerak. * Berpaling ke arah sumber suara. * Mengamati benda benda yang kemudian dipegang dijatuhkan. | dan |
| 9-12 Bulan | * Memahami perintah sederhana. * Menunjukkan reaksi saat namanya dipanggil. * Mencoba mencari benda yang disembunyikan. * Mencoba membuka atau melepas benda yang tertutup. |  |

|  |  |
| --- | --- |
| 12-18 Bulan | * Menyebutkan beberapa nama benda. * Menanyakan nama benda yang belum dikenal. * Membedakan ukuran benda (besar-kecil). * Mengenal beberapa warna primer (merah, biru, kuning). * Menyebut nama sendiri dan orang-orang yang dikenalnya. |
| 18-24 Bulan | * Mempergunakan alat permainan dengan cara semaunya. * Meniru gambar wajah orang. * Memahami konsep angka dan hitungan sederhana. * Memahami prinsip milik orang lain. |
| 2-3 Tahun | * Menyebut bagian-bagian suatu gambar (wajah orang, mobil, binatang, dan lainnya). * Memahami prinsip ukuran (besar-kecil, panjang-pendek). * Mengenal kembali bagian-bagian tubuh (lima bagian). * Mengenal tiga macam bentuk geometri, seperti lingkaran, segitiga, dan persegi empat. |
| 3-4 Tahun | * Menempatkan benda dalam urutan berdasarkan ukuran (paling kecil-paling besar) * Menemukan/mengenali bagian yang hilang dari suatu pola gambar (wajah orang, mobil, dan lainnya). * Mengekspresikan diri * Memahami perbedaan antara dua hal dari jenis yang sama (misalnya perbedaan antara buah rambutan dan pisang, perbedaan antara ayam dan kucing) |
| 4-5 Tahun | * Mengkasifikasi benda berdasarkan bentuk, warna, atau ukuran. * Menyebutkan beberapa angka dan huruf. * Menggunakan benda-benda sebagai permainan simbolik   (misalnya kursi sebagai mobil).   * Mengenal sebab-akibat tentang alam sekitar. |
| 5-6 Tahun | - Mengkasifikan benda berdasarkan fungsinya (misalnya pensil untuk menulis) |
|  | * Menunjukkan kegiatan yang bersifat eksploratif dan menyelidik. * Mencari alternatif dalam memecahkan masalah yang dihadapi dalam suatu aktivitas. * Menyusun perencanaan kegiatan yang akan dilakukan Bersama teman-teman. * Menunjukkan inisatif dan kreativitas dalam memilih tema permainan. |

Perkembangan kognitif pada anak usia dini perlu dioptimalkan. Ada beberapa alasan mengapa hal itu harus dilakukan, yaitu:

1. Agar anak mampu mengembangkan daya presepsinya berdasarkan apa yang dilihat, didengar, dan dirasakan sehingga anak memiliki pemahaman yang utuh dan komprehensif.
2. Agar anak mampu melatih ingatannya terhadap semua peristiwa dan kejadian yang pernah dialaminya.
3. Agar anak mampu mengembangkan berbagai pemikiran dalam rangka menghubungkan antara suatu peristiwa dengan peristiwa lainnya.
4. Agar anak memahami berbagai simbol yang terdapat di lingkungan sekitarnya.
5. Agar anak mampu melakukan berbagai proses penalaran, baik yang terjadi melalui proses alamiah (spontan) maupun melalui proses ilmiah (eksperimen).
6. Agar anak mampu memecahkan problematika hidup yang dihadapinya sehingga pada akhirnya ia menjadi individu yang mampu menolong dirinya sendiri.

## 2.3 Pembangkit Bilangan Acak

Pembangkit bilangan acak atau *random number generator* adalah suatu algoritma yang digunakan untuk menghasilkan urutan-urutan atau *sequence* dari angka-angka sebagai hasil dari perhitungan dengan komputer yang diketahui distribusinya sehingga angka-angka tersebut muncul secara randomdan digunakan terus-menerus (Kakiay, 2004). Defenisi tersebut dapat ditarik tiga pokok pengertian sebagai berikut :

1. Urutan (*Sequence*) Yang dimaksud dengan *sequence* adalah bahwa bilangan acak tersebut harus dapat dihasilkan secara urut dalam jumlah yang mengikuti algoritma tertentu dan sesuai dengan distribusi yang akan terjadi atau yang dikehendaki.
2. Distribusi (*Distribution*) Pengertian distribusi berhubungan distribusi probabilitas yang dipergunakan untuk meninjau atau terlibat langsung dalam penarikan bilangan acak tersebut. Pada umumnya distribusi probabilitas untuk bilangan acak ini adalah uniform variate yang dikenal dengan distribusi uniform.
3. Muncul angka-angka secara random, pengertian random disini menunjukkan bahwa algoritma tersebut akan menghasilkan suatu angka yang akan berperan dalam pemunculan angka yang keluar dalam proses di komputer. Dengan kata lain suatu angka yang diperoleh merupakan angka penentu bagi angka random berikutnya, dan demikian seterusnya. Walaupun random number ini saling berkaitan namun angka-angka yang muncul dapat berlain-lainan.

## 2.4 *Linear Congruent Method*

*Linear Congruent Method* (LCM) merupakan metode pembangkit bilangan acak yang banyak digunakan dalam program komputer. Salah satu sifat dari metode ini adalah terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan (Sulindawaty, 2011).

*Linear Congruent Method* adalah algoritma paling terkenal dan paling banyak digunakan untuk menghasilkan angka acak. Keuntungan praktis mereka adalah kecepatan, kemudahan implementasi, dan ketersediaan kode portabel, parameter dan hasil tes. *Generator congruential linear* adalah pembangkit bilangan acak klasik.

DH Lehmer pada tahun 1948 mengusulkan linier kongruen generator sebagai sumber nomor acak. Dalam generator ini, setiap nomor tunggal menentukan penggantinya dengan cara fungsi linear sederhana diikuti dengan modular reduksi. Meskipun generator ini terbatas pada kemampuannya untuk menghasilkan sangat panjang aliran angka yang tampaknya realisasi independen sebuah proses yang seragam, itu merupakan unsur dasar dalam lainnya, generator lebih memadai. Memahami sifat-sifatnya yang diperlukan dalam rangka untuk menggunakannya untuk membangun lebih baik generator (Gentle, 2003).

Bentuk dari linier congruent method sebagai berikut (Darma Nasution, 2013) :

숐௝ 숐 t ………………………………………………………(1)

Keterangan :

숐௝ Angka acak yang baru.

Angka acak yang lama atau angka acak sebelumnya.

Angka konstanta pengalian. t Angka kenaikan. Angka modulo

Untuk mengatasi terjadinya pengulangan pada periode waktu tertentu maka penentuan konstanta LCM (*a, c* dan *m*) sangat menentukan baik tidaknya bilangan acak yang diperoleh dalam arti memperoleh bilangan acak yang seakanakan tidak terjadi pengulangan. Pada *Linear Congruent Method* linier ini nilai acak akan memiliki periode penuh dan bervariatif jika memenuhi syarat sebagai berikut :

1. Konstanta *a* harus lebih besar dari √m dan biasanya dinyatakan dengan syarat :

…………………………….…….………………………(2)

௝

1. Untuk konstanta t harus berangka ganjil apabila m bernilai pangkat dua. Tidak boleh nilai kelipatan dari .
2. Untuk modulo disarankan bilangan prima sehingga mempermudah perhitungan - perhitungan di dalam komputer agar dapat berjalan dengan lancar.
3. Untuk harus merupakan angka integer.

Perbandingan terjadinya pengulangan pada LCM antara pemilihan konstanta LCM yang tidak mengikuti syarat dengan yang mengikuti syarat-syarat menetukan konstanta dapat dilakukan dengan melakukan percobaan.

Contoh membangkitkan bilangan acak sebanyak 8 kali dengan : t ௝ dan

௝ 숐 ௝ 숐 ௝ 숐 ௝ 숐 ௝ ௝

௝ 숐 ௝ ௝

௝ 숐 ௝ 숐 ௝ ௝

௝ 숐 ௝

Bilangan acak yang dibangkitkan adalah : t t t ௝t t ௝ t

Dapat dilihat pada percobaan di atas tidak terdapat perulangan secara periodik.

## 2.5 Uji Keacakan

Uji keacakan dilakukan dengan tujuan untuk mengetahui apakah runtutan id soal yang tampil pada aplikasi adalah acak atau tidak acak. Penelitian ini menggunakan salah satu dari lima uji dasar untuk keacakan. *Run Test* merupakan bagian dari statistik non parametik yang dapat digunakan untuk melakukan pengujian, apakah antar resiudal terjadi korelasi yang tinggi. Apabilan antara residual tidak terdapat hubungan korelasi, maka dapat dikatakan bahwa residual adalah acak. Dengan hipotesis sebagai dasar pengambilan keputusan adalah sebagai berikut (Ghozali, 2016)

Apabila nilai Asymp. Sig. (2-tailed) kurang dari 5% atau 0,05, maka untuk H0 ditolak dan Ha diterima. Hal tersebut berarti data residual terjadi secara tidak acak (sistematis).

Apabila nilai Asymp. Sig. (2-tailed) lebih dari 5% atau 0,05, maka untuk H0 diterima dan Ha ditolak. Hal tersebut berarti data residual terjadi secara acak (random).

## 1) Asumsi

Data yang tersedia untuk analisis terdiri atas serangkaian pengamatan, yang dicatat berdasarkan urut-urutan perolehannya, dan dapat kita kategorikan ke dalam dua kelompok yang saling eksklusif. Kita mengandaikan bahwa n = ukuran sampel total, 1 = banyaknya pengamatan kelompok pada tipe satu, dan 2 = banyaknya pengamatan pada tipe lainnya.

1. (Dua-arah)
   1. : Pola perolehan (kemunculan) kedua kelompok (tipe) pengamatan ditentukan melalui suatu proses acak.
   2. : Pola perolehan tidak acak
2. (Satu-arah)
   1. : Pola perolehan kedua kelompok pengamatan ditentukan melalui proses acak
   2. : Pola perolehan tidak acak (karena rangkaian yang ada terlalu sedikit untuk bisa dianggap kebetulan)
3. (Satu-arah)
   1. : Pola perolehan kedua kelompok pengamatan ditentukan melalui proses acak .
   2. : Pola perolehan tidak acak (karena rangkaian yang ada terlalu banyak untuk bisa dianggap kebetulan)

## 2) Statistik Uji

1. Tentukan nilai median data
2. Untuk data yang > median, beri tanda +

Untuk data yang < median, beri tanda –

Untuk data yang = median, beri tanda 0

1. Setelah data dinyatakan dalam tanda + dan -, tentukan banyaknya run dalam urutan data tersebut (urutan data tidak boleh diubah)
2. Run = banyaknya urutan data dengan tanda yang identik yang diikuti dan didahului oleh tanda yang berbeda atau tanpa tanda. contoh :

* + + = 2 run
* + - - = 3 run
* - + - + - = 5 run

e. Tentukan :

1 = banyaknya data dalam kategori tertentu misalnya +

2= banyaknya data dalam kategori lainnya, misalnya – r = banyaknya run dalam urutan

## 3) Statistik Uji

Karena hipotesis nol tidak menetapkan arah, maka dalam hal ini yang sesuai adalah uji dua-arah. Oleh sebab itu kita terlebih dahulu harus mendapatkan nilai-nilai kritis bawah serta atas untuk statistik uji. Baik bila r lebih kecil daripada atau sama dengan nilai kritis bawah, maupun bila r lebih besar daripada atau sama dengan nilai kritis atas, kita menolak hipotesis nol yang menyatakan kerandoman.

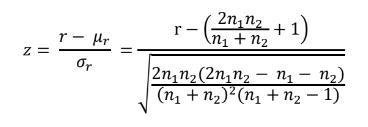
Tabel-tabel A.5 dan A.6 berturut-turut menyajikan nilai-nilai kritis bawah dan atas statistik uji ini untuk tingkat kepercayaan 0.05 dan nilai nilai 1 serta 2 hingga 20. [Tabel-tabel ini telah diadaptasikan dari tabeltabel yang disusun oleh Swed dan Eisenhart.] Untuk menentukan nilai kritis bawah, kita mengacu ke Tabel A.5 dengan 1 dan 2 yang diketahui.

Begitu pula, untuk mendapatkan nilai kritis atas, kita mengacu ke Tabel A.6 dengan 1 dan 2 yang kita miliki.

1. Untuk 1 dan 2 ≤ 20

maka 0 diterima dan 1 ditolak bila r < ra atau r > rb maka 0ditolak dan 1diterima

1. Untuk 1 atau 2 > 20

r ~ berdistribusi normal dengan rata-rata (μr) dan standard deviasi **................(2.1)**

Keterangan :

μr = rata-rata (mean)

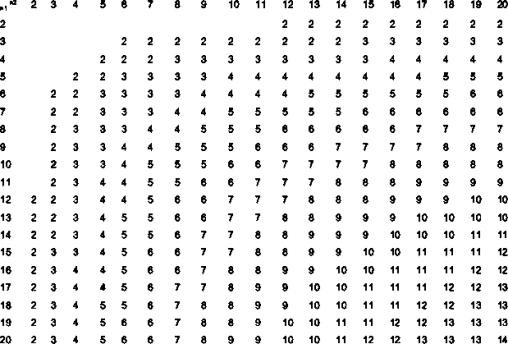
= standar deviasi z = hasil perhitungan data r = banyaknya run dalam urutan

1= banyaknya data dalam kelompok 1 / kategori 1

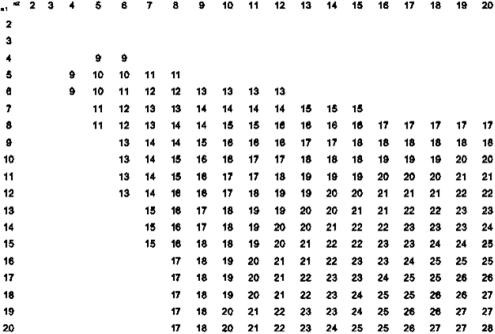
2= banyaknya data dalam kelompok 2 / kategori 2

Jika < ℎ < , maka 0 diterima, 1 ditolak sedangkan jika ℎ ≤ atau ℎ ≥ , maka 1 diterima 0 ditolak. Nilai dapat dilihat pada tabel Z kurva normal

**Tabel 2.2 Tabel A.5 (**쎠**) dengan α = 0,05**



**Tabel 2.3 Tabel A.6 ( b) dengan α = 0,05**



Contoh kasus :

Terdapat 25 sampel soal, dibangkitkan 10 soal acak menggunakan metode *Linear Congruent Method* (LCM) dengan X = 1, a = 2, c = 5, m = 25, dan diperoleh hasil 7, 19, 18, 16, 12, 4, 13, 6, 17, 14, 8. Ujilah dengan α = 0,05 apakah data tersebut mempunyai urutan yang random?

Penyelesaian :

**A) Asumsi :**

1. = urutan data merupakan urutan yang random / acak
2. = urutan data bukan merupakan urutan yang random / acak Median data :

Id soal diurutkan dari terkecil hingga terbesar :

4,6,7,12,13,15,16,17,18,19.

Median = (15+16)/2 = 15.5

Kembalika data ke urutan semula kemudian beri tanda - untuk data yang < median, tanda + untuk data yang > dari median dan 0 untuk data yang sama dengan median.

**Tabel 2.3 Hasil Pengkategorian Data**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No  Soal | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Id Soal | 7 | 19 | 18 | 16 | 12 | 4 | 13 | 6 | 17 | 15 |
| kat | - | + | + | + | - | - | - | - | + | - |

Dari tabel diperoleh :

n1 = 6 (jumlah data dengan kategori +) n2 = 4 (jumlah data dengan kategori -) r = 5

**B) Asumsi :**

Berdasarkan Tabel A.5 dan Tabel A.6 nilai ra = 2 dan rb = 9, maka memenuhi syarat ra ≤ r ≤ rb , sehingga H0 diterima dan H1 ditolak. Berarti urutan data diatas mempunyai urutan yang random

### 2.6 *Android*

*Android* merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka seßndiri untuk digunanakan oleh bermacam piranti bergerak. *Android* umum digunakan pada *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS. *Android* tidak terikat ke satu merek *handphone* saja, beberapa vendor terkenal yang sudah memakai Android antara lain Samsung, XIAOMI, VIVO, OPPO, Motorolla, Lenovo dan lain-lain (Nazaruddin, 2012).

Pada Juli 2000, Google bekerjasama dengan Android *Inc.,* perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri *Android Inc.* bekerja pada Google, diantaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android *Inc*. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Versi *android* terbaru yaitu versi 9.0 (Pie)*.*

Berikut spesifikasi android versi 9.0 (Pie). *Android Pie* adalah pembaruan selanjutnya dari *Android* 8.1 "Oreo" yang merupakan pembaruan versi ke 9 dan pembaruan urutan ke-16 dari sistem operasi *Android*, untuk versi pengembangan nya pertama kali dirilis pada Maret 2018 dan dirilis ke publik pada 6 Agustus 2018.

### 2.7 *Unified Modeling Language* (UML)

[UML](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html) [(*Unified*](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html)[*Modeling*](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html)[*Language*)](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html) adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artefact* dapat berupa model, deskripsi atau perangkat lunak dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Shalahuddin, 2013)

#### 2.7.1 *Use Case Diagram*

*Diagram Use case* adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat *user.* Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. Simbol *Use Case Diagram* dapat dilihat pada Tabel 2.

## Tabel 2.4 Simbol *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Actor* | Menspesifikasikan  himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use*  *case* |
| 2. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri  (*dependent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri  (independent) |
| 3. |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk  (*ancestor*) |
| 4. |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara  eksplisit |
| 5. |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan |
| 6. |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7. |  | *System* | Menspesifikasikan paket yang menampilkan sistem  secara terbatas |
| 8. |  | *Use case* | Deskripsi dari uraian aksiaksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor* |
| 9. |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-  elemennya (sinergi) |
| 10. |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu  sumber daya komputasi |

### 2.7.2 *Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). *Activity diagram* ini mirip dengan *flowchart diagram*. Simbol *Activity Diagram* dapat dilihat pada Tabel 3.

### Tabel 2.5 Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Activity* | Memperlihatkan bagaimana masingmasing kelas antarmuka saling berinteraksi satu sama lain. |
| 2. |  | *Action* | *State* dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3. |  | *Initial Node* | Bagaimana objek dibentuk atau diawali |
| 4. |  | *Activity*  *Final Node* | Bagaimana objek dibentuk dan diakhiri |
| 5. |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

#### 2.7.3 *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

**Tabel 2.6 Simbol S*equence Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-  informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-  informasi tentang aktifitas yang terjadi |
| 4 |  | *Message* | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy* |

#### 2.7.4 *Class Diagram*

*Class* adalah kumpulan objek-objek dengan dan yang mempunyai struktur umum, *behavior* umum, relasi umum, dan semantic/kata yang umum. *Class-class* ditentukan/ditemukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan.

**Tabel 2.7 Simbol *Class Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Association* | Apa yang menghubungkan antara objek satu dengan  objek lainnya |

|  |  |  |  |
| --- | --- | --- | --- |
| 2. |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek |
| 3. |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta  operasi yang sama |
| 4. |  | *Collaboration* | Deskripsi dari urutan aksiaksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu *actor* |
| 5. |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek |
| 6. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7. | 0..1 | *Nilai kardinitas* | Kosong atau satu. |
| 8. | 0..\* | *Nilai kardinitas* | Lebih dari sama dengan kosong. |
| 9. | 0..n | *Nilai kardinitas* | Lebih dari sama dengan n, dimana n lebih dari satu. |
| 10. | 1 | *Nilai kardinitas* | Hanya satu |
| 11. | 1..\* | *Nilai kardinitas* | Lebih dari sama dengan satu |
| 12. | 1..n | *Nilai kardinitas* | Lebih dari sama dengan satu dimana n lebih dari satu. |
| 13. | \* | *Nilai kardinitas* | Banyak atau *Many.* |
| 14. | 1 | *Nilai kardinitas* | Hanya N, dimana N lebih |
|  |  |  | dari satu. |
| 15. | n..\* | *Nilai kardinitas* | Lebih dari sama dengan N dimana N lebih dari satu. |
| 16. | n..m | *Nilai kardinitas* | Lebih dari sama dengan N dan kurang dari sama dengan M. Dimana M dan N lebih dari satu. |

#### 2.8 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. Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program/aplikasi (Oktaviani, 2018)

##### 2.8.1 *Typescript*

*Typescript* adalah bahasa pemrograman berbasis JavaScript yang menambahkan fitur *strong-typing* & konsep pemrograman OOP klasik (*class*, *interface*). Di dalam dokumentasinya, TypeScript disebut sebagai *super-set* dari JavaScript, artinya semua kode JavaScript adalah kode TypeScript juga. Bahasa pemrograman ini menawarkan *class*, *module*, dan *interface* yang membuat *developer* bisa mengembangkan aplikasi kompleks dengan lebih mudah. Hal inilah yang membedakannya dengan *javascript* (Hidayat, 2016)

TypeScript menawarkan class, module, dan interface yang membuat developer bisa mengembangkan aplikasi kompleks dengan lebih mudah diantara keunggulan dan fitur – fiturnya (Hidayat, 2016), antara lain :

1. *Support Class* dan *Module*
2. *Static Type-checking*
3. *Support* ES6 *Feature*
4. *Clear Library API Definition*
5. *Build-in Support* untuk JavaScript *Packaging*
6. Kesamaan *Syntax* untuk *Backend*
7. *Superset* dari JavaScript

##### 2.8.2 Angular-CLI

Angular-CLI merupakan generator untuk project Angular 2 dan 4 (serta versi diatasnya), dengan menggunakan bantuan Angular-CLI ini kita tidak perlu lagi *setup* banyak kebutuhan dasar seperti susunan *folder*, cara *run*, cara *build* dan lain-lain. Kesemuanya itu telah ada perintahnya (*command*) melalui Angular-CLI. Seperti halnya Ember.js yang mempunyai Ember-CLI, Vuejs dengan Vue-CLI dan React dengan “*create-react-app*” maka seperti itu pula fungsi Angular-CLI bagi Angular (Maulana, 2017).

#### 2.9 *Database*

Sistem *database* adalah suatu sistem penyusunan dan pengelolaan data dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan (Ahmad, 2007)

*Database* SQL (*Structure Query Language*) merupakan database relasional atau database yang terstruktur. Jadi, SQL database ini merupakan penyimpanan data yang menggunakan *relation* (hubungan) dan biasa disebut sebagai tabel. Artinya *Sql database ini* dapat menemukan data yang di carinya menggunakan karakteristik umum yang terdapat pada penyimpanannya dan hasil pencariannya disebut dengan skema (Kadir, 2015).

#### 2.10 SQLite

*Database* SQLite adalah solusi penyimpanan yang baik jika anda memiliki data terstruktur yang perlu diakses dan disimpan secara persisten serta sering ditelusuri dan diubah. Anda juga bisa menggunakan SQLite sebagai media penyimpanan utama untuk data aplikasi atau pengguna, atau anda juga bisa menggunakannya untuk proses caching serta menyediakan data yang diambil dari cloud.

Jika anda bisa menyatakan data berupa baris dan kolom, pertimbangkan untuk memakai database SQLite. Jika anda menggunakan database SQLite, yang dinyatakan sebagai objek SQLiteDatabase adalah semua interaksi dengan database adalah melalui instance dari kelas SQLiteOpenHelper yang akan mengeksekusi permintaan dan pengelolaan database (Ridwansyah, 2018).

#### 2.11 *Android SDK (Software Development Kid)*

Sesuai dengan namanya merupakan kumpulan tools-tools yang di perlukan untuk mengembangkan aplikasi Android. Seperti pada *java* yang dapat dijalankan pada *multiplatform*, SDK Android pun tersedia untuk beberapa platform seperti Windows, Mac OS dan Linux. Adapun aplikasi SDK yang didalamnya terdapat beberapa folder sebagai berikut (Mulyadi, 2010):

1. *Tools*
2. Berisikan tools-tools yang diperlukan dalam membangun aplikasi Android seperti emulator, *Android Virtual Device (AVD), SDK Manager, ddms* dan banyak lagi untuk memudahkan membangun aplikasi Android.
3. *Platforms*

Direktori ini ditujukkan untuk meletakkan versi platform-platform Android yang telah pengguna download dari *SDK Manager.* Seperti Android

1.5, 1.6, 2.0 dan seterusnya.

1. *Add-ons*

Direktori ini nantinya berisikan *tools* tambahan untuk Android SDK, yang fungsinya untuk menambahkan *library* yang akan digunakan untuk pengembangan aplikasi. Seperti jika ingin mengembangkan aplikasi menggunakan *map* dari google maka harus memerlukan platform Android yang sudah terdapat *library map.*

#### 2.12 Pengujian *Black Box*

Pengujian *black box* merupakan pendekatan komplementer dari teknik *white box*, karena pengujian *black box* diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik *white box*. Pengujian *black box* berfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program. Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Pengujian *black box* berusaha menemukan kesalahan dalam kategori (Rouf, 2012):

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface.*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Berbeda dengan pengujian *white box*, pengujian *black box* cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *black box* harus dapat menjawab pertanyaan sebagai berikut :

1. Bagaimana validasi fungsional diuji ?
2. Masukkan apa yang akan membuat kasus pengujian menjadi lebih baik ?
3. Apakah sistem akan sangat sensitif terhadap masukkan tertentu ?
4. Apakah aplikasi dapat bekerja disistem operasi yang berbeda ?

**BAB II**

**LANDASAN TEORI**

## 2.1 *Game*

Berdasarkan kamus, *game* berasal dari kata bahasa inggris yang berarti permainan. Banyak teori yang mengungkapkan tentang pengertian *game*. *Game* merupakan suatu sistem atau program dimana satu atau lebih pemain mengambil keputusan melalui kendali pada obyek didalam *game* untuk suatu tujuan tertentu. Teori lain mengatakan bahwa *game* merupakan salah satu media hiburan yang menjadi pilihan masyarakat untuk menghilangkan kejenuhan atau untuk mengisi waktu luang. Selain sebagai media hiburan, *game* juga dapat meningkatkan perkembangan otak seseorang, contohnya adalah permainan catur yang dapat meningkatkan konsentrasi otak (Yandi, 2014).

Berdasarkan pengertian tersebut dapat dikatakan bahwa *game* atau permainan merupakan suatu media yang digunakan untuk tujuan hiburan dimana pemainnya melakukan suatu kegiatan guna mencapai tujuan tertentu yang biasanya dapat dilakukan secara mandiri atau *single player* maupun lebih dari satu pemain atau *multiplayer*. Proses pembuatan *game* biasanya pembuat *game* memiliki suatu tujuan tertentu yang disesuaikan dengan kebutuhan dan keinginan pembuat *game*. Tujuan pembuatan *game* antara lain (Yandi, 2014) : a. Sebagai hiburan (*Entertainment*)

1. Melatih ketangkasan (*Expand Skill*)
2. Mendidik (*Education*)
3. Menyampaikan pesan (*Embed Messages*)

### 2.1.1 Jenis *Game*

Jenis *game* umumnya disebut juga dengan istilah *genre game* yang dapat diklasifikasikan sebagai berikut (Sagala, 2009) :

a. *Maze Game*

Jenis game ini adalah jenis *game* yang paling awal muncul. Secara sederhana kita hanya mengitari *maze* (lorong-lorong yang berhubungan) dan memakan beberapa item untuk menambah tenaga atau kekebalan. Kita juga tentunya memiliki musuh yang mengejar kita. Ketika kita mendapatkan kekebalan kita bisa berbalik mengejar mereka mereka. Mode permainan inilaiS yang menjadi dasar dari permainan 3D sekarang. Contoh maze *game* ialah *Digger, Pacman, Doom, Ultimate Doom, Quake.* b. *Board Game*

Jenis *game* ini sama dengan *game* board tradisional, seperti *Monopoly*. Sampai saat ini tidak ada variasi yang memunculkan gameplay ataupun perubahan desain dari versi tradisional ke versi elektronik. Versi elektronik benar-benar hanya memindahkan versi tradisional ke layar komputer. Umumnya *game* ini lebih menekankan kepada kemampuan komputer menjadi lawan tanding dari pemain. Contoh *game* jenis ini ialah *Chess, Monopoly, Scrabble*. c. *Card Game*

Hampir sama dengan *board game*, genre ini tidak memberikan perubahan berarti dari game tradisional yang sejenis. Contohnya, game *Solitaire dan Hearts,* versi asli dan elektroniknya nyaris tidak ada bedanya. Variasinya adalah kemampuan *multiplayer* dan tampilan yang lebih bervarisi dari versi tradisional. *Game* ini termasuk game yang muncul pada awal *game* komputer seperti genre maze dan board *game*. Contohnya, *Hearts, Spider, Balckjack, Poker, Solitaire*. d. *Quiz Game*

Jenis ini juga agak jarang di Indonesia. Salah satu yang umum dikenal adalah *game* kuis *Who Wants To Be Milionare* sebuah *game* dengan nama yang sama dari acara kuis televisi. *Game* ini sederhana dalam cara bermain. Kita hanya perlu memilih jawaban yang benar dari beberapa pilihan jawaban yang benar dari beberapa pilihan jawaban. Biasanya pertanyaan yang diberikan memang memiliki topik tertentu. Contoh lainnya ialah, *Clasrooom Milionare, Deal or No Deal*.

e. *Puzzle Game*

*Game* ini memberikan tantangan kepada pemainnya dengan cara menjatuhkan sesuatu dari sisi sebelah atas ke bawah. Pemain harus menyusunnya sedemikian rupa dan tidak ada yang tersisa ketika susunan diatas sudah akan dibuat. Susunan ini dilakukan secepat dan sebaik mungkin. Semakin lama akan semakin cepat dan semakin banyak objek yang jatuh. Dalam perkembangan, jenis ini membebaskan cara bermainnya. Misalnya dengan *user* bebas meletakkan objek ke suatu tempat dengan tujuan tertentu. Contoh *game* ini ialah *Magic Inlay, Adventure Inlay, Tetris, Chip Challenge.* f. *Sport Game*

*Game* ini sama dengan *game* endutaiment. Genre ini hanya berdasarkan jenisnya, bukan berdasarkan teknologi atau spesifikasi teknis apapun. Selama *game* itu mengetengahkan genre olahraga maka disebut *genre sport* tidak peduli apakah *game* ini menggunakan gaya arcade 2D atau 3D. Contoh : *Winning Eleven, Championship Manager, FIFA, NBA, Virtual Tennis.* g. *Racing Game*

*Game* balapan, *game* ini memberikan permainan lomba kecepatan dari kendaraan yang dimainkan oleh pemain. Terkadang didalam arena, terkadang diluar arena balap. Contoh *game* ini yaitu, *Need For Speed, NFS Underground, Driver, Taxi.*

h. *Simulation Game*

Disini kita sebagai pemain membangun secara simulasi sebuah kota, negara atau koloni. Kita mengatur berbagai sumber daya dan menentukan berbagai keputusan yang kita inginkan dalam proses pembangunan yang sedang terjadi. Disamping itu pula terdapat *sub-genre* dari *simulation game* seperti *flight simulator* dan *technical simulator.* i. *Turn Based Strategy Game*

*Game* ini memerlukan strategi dari pemain untuk memenangkan permainan. Pemain melakukan gerakan setelah pemain lain melakukannya, jadi saling bergantian. Bisa dibilang mirip dengan catur, tetapi dengan variasi gerakan dan efek yang jauh lebih banyak. Contoh: *Empire, Civilization, Heroes of Might and Magic.*

1. *Real Time Strategy game*

Jika pada *Turn Based Strategy game* kita perlu menunggu pemain lain menyerang, pada *Real Time Strategy game* ini kita tidak perlu menunggu. Pemain yang tercepatlah yang besar kemungkinannya untuk menang. Pada jenis *game* ini kita harus melakukan beberapa gerakan sesuai dengan strategi kita. Contoh *game* jenis ini ialah *Warcraft, Starcraft, Commandos, Command and Conquer.*

1. *Role Playing Game*

*Genre game* ini kita akan berperan menjadi sebuah karakter. Kita akan menjalankan peran kita ini dengan berbagai atribut, seperti kesehatan, intelegensi, kekuatan, dan keahlian. Salah satu *game* yang terkenal dengan RPG pada masa awal adalah Ultima. Kini *genre* ini berkembang menjadi beberapa jenis variasi RPG seperti action *Role Playing Game.*

Berdasarkan jenis *game* tersebut, jenis *game simulation game, sport game, role playing game, real time strategy dan racing* yang paling popular dari kalangan anak-anak hingga remaja di industri *game* baik *mobile* maupun komputer. Untuk *game* kategori *puzzle dan quiz* lebih ditujukan kepada anakanak sebagai media pengenalan objek dan edukasi (Sagala, 2009).

### 2.1.2 *Game* Edukasi

Penerapan *game* edukasi bermula dari perkembangan *video game* yang sangat pesat dan menjadikannya sebagai media yang efektif, interaktif dan banyak dikembangkan di perusahaan industri. Melihat kepopuleran game tersebut, para pendidik berpikir bahwa komponen rancangan *game* dapat digunakan sebagai pengantar materi pembelajaran serta dapat diterapkan pada kurikulum pendidikan. *Game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan. Kriteria yang harus dimiliki dari *game* edukasi sebagai berikut (Hurd, 2009):

1. Nilai Keseluruhan

Nilai keseluruhan berhubungan dengan semua hal yang terdapat dalam *game* edukasi misalnya cara bermain, *game* edukasi dapat dimainkan kembali dan biaya pembuatan harus diperhatikan dengan baik.

1. Kegunaan (*Usability*)

Kegunaan berhubungan dengan seberapa baik *game* edukasi dapat memberikan pengetahuan bagi penggunanya.

1. Keakuratan (*Accuracy*)

Keakuratan berhubungan dengan kesesuaian konten yang terdapat dalam *game* edukasi pada proses pembelajaran, sehingga dapat menambah ilmu pengetahuan penggunanya.

1. Kelayakan

Kelayakan berhubungan dengan bagaimana konten dan desain *game* edukasi disesuaikan dengan sasaran pengguna. Suatu *game* edukasi dikatakan layak jika tujuan dari *game* edukasi yaitu untuk memberikan keahlian dan pengetahuan khusus bagi pengguna tercapai.

1. Hubungan (*Relationship*)

Kriteria ini berhubungan dengan bagaiman suatu konten yang ada dalam *game* edukasi. Hal ini bertujuan agar *game* edukasi yang dibuat dapat dimainkan dengan baik sesuai dengan umur dan karakteristik pengguna.

1. Tujuan

Kriteria ini berhubungan dengan apa yang didapat dari memainkan game edukasi. *Game* edukasi harus dapat memberikan manfaat bagi penggunanya sehingga konten yang ada dalam *game* edukasi harus jelas, layak dan bersifat objektif.

1. Umpan Balik

*Game* edukasi harus memberikan umpan balik yang bersifat positif misalnya pemberian efek suara, indikasi benar atau salah, keterangan setelah menyelesaikan *game* dan sebagainya.

1. Kesenangan

Kriteria ini berhubungan dengan bagaimana pengguna dapat menikmati pemainan yang ada di dalam *game* edukasi. *Game* edukasi harus dapat memberikan kesenangan dan ketertarikan bagi pengguna. *Game* edukasi diharapkan dapat meningkatkan ilmu pengetahuan dan keahlian penggunanya.

## 2.2 Anak Usia Dini

### 2.2.1 Pengertian Anak Usia Dini

Anak adalah manusia kecil yang memiliki potensi yang masih harus dikembangkan. Anak memiliki karakteristik tertentu yang khas dan tidak sama dengan orang dewasa serta akan berkembang menjadi manusia dewasa seutuhnya. Menurut Undang-undang Sisdiknas tahun 2003 "Anak usia dini adalah anak yang berada pada rentang usia 0-6 tahun”. Anak memiliki karakteristik tertentu yang khas dan tidak sama dengan orang dewasa. Anak selalu aktif, dinamis, antusias dan ingin tahu terhadap apa yang dilihat dan didenganya, seolah-olah tidak pernah berhenti belajar (Kementerian Pendidikan dan Kebudayaan, 2003). Mansur mengungkapkan bahwa "Anak usia dini adalah kelompok anak yang berada dalam proses pertumbuhan dan perkembangan yang khusus sesuai dengan tingkat pertumbuhan dan perkembangan” (Mansur, 2005). Sejalan dengan hal tersebut, Hartati mengungkapkan bahwa "Anak usia dini adalah seorang manusia atau individu yang memiliki pola perkembangan dan kebutuhan tertentu yang berbeda dengan orang dewasa” (Hartati, 2005). Berdasarkan ungkapan di atas maka dapat disimpulkan bahwa, anak usia dini merupakan individu berusia 0-6 tahun yang memiliki ciri khas unik dan sedang dalam tahap pertumbuhan serta perkembangan, baik fisik maupun mental. Pendidikan anak usia dini akan membantu proses pertumbuhan dan perkembangan anak menjadi manusia yang lebih baik menuju kematangan.

### 2.2.2 Karakteristik Anak Usia Dini

Anak usia dini memiliki karakteristik yang berbeda dengan orang dewasa, beberapa karakteristik anak usia dini yaitu (Hartati, 2005): a. Anak bersifat egosentris

1. Anak memiliki rasa ingin tahu yang besar
2. Anak adalah mahluk sosial
3. Anak bersifat unik
4. Anak umumnya kaya dengan fantasi
5. Anak memiliki daya konsentrasi yang pendek
6. Anak merupakan masa belajar yang paling potensial

Sejalan dengan ungkapan tersebut menjelaskan bahwa anak usia dini memiliki karakteristik yaitu : a. Bersifat egosentris naif

1. Mempunyai relasi sosial dengan benda-benda dan manusia yang sifatnya sederhana dan primitif
2. Ada kesatuan jasmani dan rohani yang hampir tidak terpisahkan sebagai satu totalitas
3. Sikap hidup yang fungsionis

Berdasarkan beberapa ungakapan diatas dapat disimpulkan bahwa anak memiliki karakteristik yang jauh berbeda dengan orang dewasa. Anak adalah sosok individu yang sangat aktif, dinamis antusias dan selalu ingin tahu terhadap apa yang dilihat dan didengarnya.

### 2.2.3 Perkembangan Anak Usia Dini

Perkembangan anak usia dini memiliki beberapa aspek yaitu aspek fisikmotorik, kognitif, bahasa, seni dan sosial-emosional. Aspek- aspek tersebut tidak dapat berkembang sendiri-sendiri, melainkan aspek-aspek tersebut saling berkaitan.apabila salah satu aspek tidak dapat berkembang dengan baik maka aspek-aspek yang lainnya juga terhambat perkembangannya. Dibawah ini adalah aspek-aspek tersebut (Hartati, 2005):

a. Perkembangan Fisik-Motorik

Aspek perkembangan fisik-motorik mencakup pertumbuhan fisik pada setiap anak tidak selalu sama, ada beberapa anak yang mengalami pertumbuhan secara cepat, dan ada pula yang mengalami kelambatan. Pada usia yang sama juga kadang kita temukan satu anak memiliki badan yang tinggi dan anak lainnya lebih pendek. Pada masa usia dini, pertumbuhan tinggi dan berat badan relatif seimbang, tetapi secara bertahap tubuh anak akan mengalami perubahan.

Apabila dimasa bayi anak memiliki penampilan yang gemuk maka secara perlahan tubuhnya akan berubah menjadi lebih langsing, sedangkan kaki dan tangannya mulai memanjang. Perkembangan motorik anak juga sudah berkembang baik. Jika pada usia 1 tahun anak ada yang belum terampil berjalan, maka pada usia 2,5 tahun anak umumnya sudah dapat berlari, melompat, menendang bola, dan memanjat. Setiap gerakannya sudah selaras dengan kebutuhan atau minatnya. Pada masa ini ditandai dengan kelebihan gerak atau aktivitas. Anak cenderung menunjukkan gerakan-gerakan motorik yang cukup gesit dan lincah. Oleh karena itu, usia ini merupakan masa ideal untuk belajar keterampilan yang berkaitan dengan motorik, seperti menulis, menggambar, melukis, berenang, main bola dan atletik. b. Perkembangan Kognitif

Perkembangan kognitif menyangkut perkembangan berfikir dan bagaimana kegiatan berfikir itu bekerja. Dalam kehidupannya, mungkin saja anak dihadapkan pada persoalan yang menuntut adanya pemecahan. Menyelesaikan suatu persoalan merupakan langkah yang lebih kompleks pada diri anak. Sebelum anak mampu menyelesaikan persoalan, anak perlu memiliki kemampuan untuk mencari cara penyelesainnya. Faktor kognitif mempunyai peranan yang penting bagi keberhasilan anak dalam belajar karena sebagian besar aktifitas dalam belajar selalu berhubungan dengan masalah mengingat dan berfikir. Perkembangan struktur kognitif berlangsung menurut urutan yang sama bagi semua anak. c. Perkembangan Bahasa

Bahasa merupakan sarana berkomunikasi dengan orang lain. Dalam pengertian ini tercakup semua cara untuk berkomunikasi, dimana pikiran dan perasaan dinyatakan dalam bentuk tulisan, lisan, isyarat atau gerak dengan menggunakan kata-kata, kalimat bunyi, lambang, gambar atau lukisan. Menurut Miller, bahasa adalah suatu urutan kata-kata, bahasa juga dapat digunakan untuk menyampaikan informasi mengenai tempat atau waktu yang berbeda. d. Perkembangan Sosial Emosional

Pada usia dini, emosi anak mulai matang. Anak mulai menyadari akibatakibat dari tampilan emosinya. Anak mulai memahami perasaan orang lain, misalnya bagaimana perasaan orang lain apabila disakiti maka anak belajar mengendalikan emosinya. e. Perkembangan Seni

Melalui seni anak dapat mengembangkan beberapa aspek perkembangan lainnya seperti menyanyi sambil belajar huruf dan angka untuk membantu mengembangkan aspek perkembangan kognitif atau menggunting, menggambar dan menari untuk mengembangkan aspek perkembangan kognitif, fisik, dan motorik anak. Kemampuan anak usia dini untuk merasakan dan melakukan berbagai keterampilan atau kemampuan seninya dapat ditimbulkan dan dikembangkan sejak dini melalui pelatihan dan bimbingan yang terarah sambil disesuaikan dengan karakteristik belajar anak usia dini yaitu bermain.

### 2.2.4 Program Pendidikan Anak Usia Dini

Anak usia dini yang biasa disingkat dengan AUD adalah anak yang berusia 0 hingga 6 tahun sebelum memasuki masa sekolah dasar. Tahapan yang dilalui anak pada masa usia dini adalah masa bayi dari usia lahir sampai dengan 12 bulan, masa kanak-kanak dari usia 1 tahun hingga 3 tahun. Selanjutnya masa pra sekolah dari usia 3 tahun sampai dengan usia 6 tahun. Berbagai aspek pertumbuhan dan perkembangan yang meliputi aspek pertumbuhan fisik, perkembangan kognitif, emosi, sosial, bahasa serta perkembangan moral dan agama.

Bidang garapan pendidikan anak usia dini meliputi tiga jenis yang masingmasing dibagi menurut usia anak. Lembaga pendidikan yang dimaksud terdiri atas:

Pendidikan Keluarga yang dikenal dengan nama Taman Penitipan Anak (TPA), Kelompok Bermain (KB) atau *play group*, dan Taman Kanak-kanak (TK).

Pendidikan Keluarga untuk anak usia 0 sampai 2 tahun, merupakan lingkungan yang pertama dan utama bagi anak sebab pendidikan keluarga merupakan fondasi bagi anak untuk membangun struktur kepribadian selanjutnya.

Taman Pengasuhan Anak (TPA) untuk anak usia 3 bulan sampai dengan 3 tahun. TPA adalah lembaga kesejahteraan sosial yang memberikan layanan pengganti berupa asuhan, perawatan dan pendidikan bagi anak balita selama anak tersebut ditinggal bekerja oleh orang tuanya. Hal ini untuk menghidarkan anak dari kemungkinan terlantar pertumbuhan dan perkembangan jasmani, rohani dan sosialnya. Kelompok Bermain (KB) atau *play group* untuk anak usia 3 sampai 4 tahun, yang merupakan tempat bermain dan belajar bagi anak-anak sebelum memasuki TK yang bertujuan mengembangkan seluruh aspek fisik, mental, emosi dan sosial anak. Isi program merupakan penjabaran dari visi dan misi serta tujuan kelompok bermain.

Taman Kanak-kanak (TK) adalah program untuk anak usia 4 sampai 6 tahun, yang merupakan jenjang pendidikan setelah kelompok bermain/play group sebelum anak masuk sekolah dasar. Pada saat ini taman Kanak-kanak bukan jenjang pendidikan wajib, dan tidak termasuk dalam program wajib belajar pendidikan dasar. Meskipun demikian, keberadaannya telah memberikan sesuatu yang cukup berarti bagi penyiapan anak usia dini memasuki pendidikan dasar.

Bina Keluarga Balita (BKB) adalah suatu kegiatan yang bertujuan memberikan pengetahuan dan keterampilan kepada orang tua dan anggota keluarga lainnya mengenai bagaimana mendidik, mengasuh dan memantau pertumbuhan dan perkembangan anak. Program ini diperuntukkan bagi ibu-ibu yang memiliki anak usi dini dan termasuk dalam kategori keluarga berpenghasilan rendah). Tujuan BKB adalah agar orang tua memiliki konsep diri yang sehat, terjadi peningkatan pengetahuan dan keterampilan dalam mengasuh dan membina anak serta mampu menerapkan pola asuh yang tepat sejak dini (Mulyasa, 2012:

55).

Undang-undang Nomor 20 tahun 2003 tentang Sistem Pendidikan Nasional, PAUD memiliki bagian tersendiri, yaitu pada Bagian Ketujuh pasal 28 dinyatakan dalam 6 ayat yang meliputi: (1) Pendidikan anak usia dini diselenggarakan sebelum jenjang pendidikan dasar. (2) Pendidikan anak usia dini dapat diselenggarakan melalui jalur pendidikan formal, nonformal, dan/atau informal. (3) Pendidikan anak usia dini pada jalur pendidikan formal berbentuk Taman Kanak-Kanak (TK), Raudatul Athfal (RA), atau bentuk lain yang sederajat.

(4) Pendidikan anak usia dini pada jalur pendidikan nonformal berbentuk Kelompok Bermain (KB), Taman Penitipan Anak (TPA), atau bentuk lain yang sederajat. (5) Pendidikan anak usia dini pada jalur pendidikan informal berbentuk pendidikan keluarga atau pendidikan yang diselenggarakan oleh lingkungan. (6) Ketentuan mengenai pendidikan anak usia dini sebagaimana dimaksud pada ayat

(1), ayat (2), ayat (3), dan ayat (4) diatur lebih lanjut dengan Peraturan Pemerintah. Taman Kanak-kanak (TK) sebagaimana dinyatakan dalam Undang-undang Republik Indonesia Nomor 20 tahun 2003 pasal 28 ayat 3 merupakan pendidikan anak usia dini pada jalur pendidikan formal yang bertujuan membantu anak didik mengembangkan berbagai potensi baik psikis dan fisik yang meliputi moral dan nilai agama, sosial, emosional, kemandirian, kognitif, bahasa, fisik/motorik dan seni untuk siap memasuki sekolah dasar. Selanjutnya dalam Peraturan Pemerintah Nomor 27 tahun 1990 Bab I pasal 1 ayat (2) dinyatakan bahwa “Taman KanakKanak adalah satu bentuk pendidikan prasekolah yang menyediakan program pendidikan dini bagi anak usia dini bagi anak usia empat sampai memasuki pendidikan dasar”.

Menurut Petunjuk Teknis Penyelenggaraan Taman Kanak-kanak, pengertian Taman Kanak-Kanak yang selanjutnya disingkat TK adalah salah satu bentuk satuan PAUD pada jalur pendidikan formal yang menyelenggarakan program pendidikan bagi anak berusia 4 tahun sampai dengan 6 tahun dengan prioritas usia 5 dan 6 tahun (Kemendikbud, 2015: 3).

Perkembangan pada anak usia dini terkait dengan perubahan psikis pada diri mereka, yang mencangkup aspek kognitif, bahasa, social, dan emosi, serta agama dan moral. Perkembangan kognitif pada anak usia dini terkait dengan pengetahuannya, yaitu semua proses psikologis yang berhubungan dengan bagaimana anak mempelajari dan memikirkan lingkungannya (Rosda, 2009). Kata kognitif berasal dari *cognition* yang sinonimnya adalah *knowing* yang berarti mengetahui.

Ada beberapa teori yang menjelaskan bagaimana perkembangan kognitif pada anak usia dini. Pertama, teori tahap perkembangan kognitif menurut Jean Piaget mengungkapkan bahwa manusia dalam hidupnya pasti melalui empat tahap perkembangan kognitif, di mana masing-masing tahap terkait dengan usia dan terdiri dari cara berpikir yang khas atau berbeda. Keempat tahap tersebut antara lain tahap *sensorimotor*, pra operasional konkret, dan tahap operasional formal.

Anak usia dini berada pada tahap *sensorimotor* dan tahap pra operasional. Anak yang berusia 0 hingga 2 tahun berada pada tahap *sensorimotor*. Tahap ini ditandai dengan kemampuan anak dalam melakukan gerak refleks sederhana (lahir sampai 1 bulan), belajar melakukan gerakan secara berulang-ulang (1-4 bulan), mulai menirukan gerakan sederhana (4-8 bulan), melihat suatu benda dan terdorong untuk menggenggamnya (8-12 bulan), mulai memiliki rasa ingin tahu dan mulai muncul minat (12-18 bulan), dan mampu menggunakan simbol-simbol sederhana (18-24 bulan).

Kemudian pada tahap pra operasional (2-4 tahun) merupakan tahap awal pembentukan konsep secara stabil. Penalaran mental mulai muncul, egosentrisme mulai kuat dan kemudian lemah, serta keyakinan terhadap hal yang magis terbentuk.

Pemberian stimulasi dalam optimalisasi perkembangan kognitif pada anak usia dini memiliki kemampuan berikut ini :

## Tabel 2.1 Optimalisasi Perkembangan Kognitif pada Anak Usia Dini

|  |  |  |
| --- | --- | --- |
| **USIA** | **KEMAMPUAN KOGNITIF** | |
| 0-3 Bulan | * Mampu membedakan apa yang diinginkan (ASI, susu botol, atau kempong/*pacifier*). * Berhenti menangis setelah digendong atau diberi susu. | dari |
| 3-6 Bulan | * Memperlihatkan dan memilih permainan yang diinginkan. * Mengelurkan kedua tangan untuk digendong. |  |
| 6-9 Bulan | * Mengamati benda-benda yang bergerak. * Berpaling ke arah sumber suara. * Mengamati benda benda yang kemudian dipegang dijatuhkan. | dan |
| 9-12 Bulan | * Memahami perintah sederhana. * Menunjukkan reaksi saat namanya dipanggil. * Mencoba mencari benda yang disembunyikan. * Mencoba membuka atau melepas benda yang tertutup. |  |
| 12-18 Bulan | * Menyebutkan beberapa nama benda. * Menanyakan nama benda yang belum dikenal. * Membedakan ukuran benda (besar-kecil). * Mengenal beberapa warna primer (merah, biru, kuning). * Menyebut nama sendiri dan orang-orang yang dikenalnya. |  |
| 18-24 Bulan | * Mempergunakan alat permainan dengan cara semaunya. * Meniru gambar wajah orang. * Memahami konsep angka dan hitungan sederhana. * Memahami prinsip milik orang lain. |  |
| 2-3 Tahun | * Menyebut bagian-bagian suatu gambar (wajah orang, mobil, binatang, dan lainnya). * Memahami prinsip ukuran (besar-kecil, panjang-pendek). * Mengenal kembali bagian-bagian tubuh (lima bagian). * Mengenal tiga macam bentuk geometri, seperti lingkaran, segitiga, dan persegi empat. | |
| 3-4 Tahun | * Menempatkan benda dalam urutan berdasarkan ukuran (paling kecil-paling besar) * Menemukan/mengenali bagian yang hilang dari suatu pola gambar (wajah orang, mobil, dan lainnya). * Mengekspresikan diri * Memahami perbedaan antara dua hal dari jenis yang sama (misalnya perbedaan antara buah rambutan dan pisang, perbedaan antara ayam dan kucing) | |
| 4-5 Tahun | * Mengkasifikasi benda berdasarkan bentuk, warna, atau ukuran. * Menyebutkan beberapa angka dan huruf. * Menggunakan benda-benda sebagai permainan simbolik   (misalnya kursi sebagai mobil).   * Mengenal sebab-akibat tentang alam sekitar. | |
| 5-6 Tahun | * Mengkasifikan benda berdasarkan fungsinya (misalnya pensil untuk menulis) * Menunjukkan kegiatan yang bersifat eksploratif dan menyelidik. * Mencari alternatif dalam memecahkan masalah yang dihadapi dalam suatu aktivitas. * Menyusun perencanaan kegiatan yang akan dilakukan Bersama teman-teman. * Menunjukkan inisatif dan kreativitas dalam memilih tema permainan. | |

Sumber : Kementerian Pendidikan dan Kebudayaan, 2003

Perkembangan kognitif pada anak usia dini perlu dioptimalkan. Ada beberapa alasan mengapa hal itu harus dilakukan, yaitu:

1. Agar anak mampu mengembangkan daya presepsinya berdasarkan apa yang dilihat, didengar, dan dirasakan sehingga anak memiliki pemahaman yang utuh dan komprehensif.
2. Agar anak mampu melatih ingatannya terhadap semua peristiwa dan kejadian yang pernah dialaminya.
3. Agar anak mampu mengembangkan berbagai pemikiran dalam rangka menghubungkan antara suatu peristiwa dengan peristiwa lainnya.
4. Agar anak memahami berbagai simbol yang terdapat di lingkungan sekitarnya.
5. Agar anak mampu melakukan berbagai proses penalaran, baik yang terjadi melalui proses alamiah (spontan) maupun melalui proses ilmiah (eksperimen).
6. Agar anak mampu memecahkan problematika hidup yang dihadapinya sehingga pada akhirnya ia menjadi individu yang mampu menolong dirinya sendiri.

### 2.3 Pembangkit Bilangan Acak

Pembangkit bilangan acak atau *random number generator* adalah suatu algoritma yang digunakan untuk menghasilkan urutan-urutan atau *sequence* dari angka-angka sebagai hasil dari perhitungan dengan komputer yang diketahui distribusinya sehingga angka-angka tersebut muncul secara randomdan digunakan terus-menerus (Kakiay, 2004). Defenisi tersebut dapat ditarik tiga pokok pengertian sebagai berikut :

1. Urutan (*Sequence*) Yang dimaksud dengan *sequence* adalah bahwa bilangan acak tersebut harus dapat dihasilkan secara urut dalam jumlah yang mengikuti algoritma tertentu dan sesuai dengan distribusi yang akan terjadi atau yang dikehendaki.
2. Distribusi (*Distribution*) Pengertian distribusi berhubungan distribusi probabilitas yang dipergunakan untuk meninjau atau terlibat langsung dalam penarikan bilangan acak tersebut. Pada umumnya distribusi probabilitas untuk bilangan acak ini adalah uniform variate yang dikenal dengan distribusi uniform.
3. Muncul angka-angka secara random, pengertian random disini menunjukkan bahwa algoritma tersebut akan menghasilkan suatu angka yang akan berperan dalam pemunculan angka yang keluar dalam proses di komputer. Dengan kata lain suatu angka yang diperoleh merupakan angka penentu bagi angka random berikutnya, dan demikian seterusnya. Walaupun random number ini saling berkaitan namun angka-angka yang muncul dapat berlain-lainan.

### 2.4 *Linear Congruent Method*

*Linear Congruent Method* (LCM) merupakan metode pembangkit bilangan acak yang banyak digunakan dalam program komputer. Salah satu sifat dari metode ini adalah terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan (Sulindawaty, 2011).

*Linear Congruent Method* adalah algoritma paling terkenal dan paling banyak digunakan untuk menghasilkan angka acak. Keuntungan praktis mereka adalah kecepatan, kemudahan implementasi, dan ketersediaan kode portabel, parameter dan hasil tes. *Generator congruential linear* adalah pembangkit bilangan acak klasik.

DH Lehmer pada tahun 1948 mengusulkan linier kongruen generator sebagai sumber nomor acak. Dalam generator ini, setiap nomor tunggal menentukan penggantinya dengan cara fungsi linear sederhana diikuti dengan modular reduksi. Meskipun generator ini terbatas pada kemampuannya untuk menghasilkan sangat panjang aliran angka yang tampaknya realisasi independen sebuah proses yang seragam, itu merupakan unsur dasar dalam lainnya, generator lebih memadai. Memahami sifat-sifatnya yang diperlukan dalam rangka untuk menggunakannya untuk membangun lebih baik generator (Gentle, 2003).

Bentuk dari linier congruent method sebagai berikut (Darma Nasution, 2013) :

쑸⁘ 쑸 t ………………………………………………………(1)

Keterangan :

쑸⁘ Angka acak yang baru.

Angka acak yang lama atau angka acak sebelumnya.

Angka konstanta pengalian. t Angka kenaikan. Angka modulo

Untuk mengatasi terjadinya pengulangan pada periode waktu tertentu maka penentuan konstanta LCM (*a, c* dan *m*) sangat menentukan baik tidaknya bilangan acak yang diperoleh dalam arti memperoleh bilangan acak yang seakanakan tidak terjadi pengulangan. Pada *Linear Congruent Method* linier ini nilai acak akan memiliki periode penuh dan bervariatif jika memenuhi syarat sebagai berikut :

1. Konstanta *a* harus lebih besar dari √m dan biasanya dinyatakan dengan syarat :

…………………………….…….………………………(2)

⁘께께

1. Untuk konstanta t harus berangka ganjil apabila m bernilai pangkat dua. Tidak boleh nilai kelipatan dari .
2. Untuk modulo disarankan bilangan prima sehingga mempermudah perhitungan - perhitungan di dalam komputer agar dapat berjalan dengan lancar.
3. Untuk 께 harus merupakan angka integer.

Perbandingan terjadinya pengulangan pada LCM antara pemilihan konstanta LCM yang tidak mengikuti syarat dengan yang mengikuti syarat-syarat menetukan konstanta dapat dilakukan dengan melakukan percobaan.

Contoh membangkitkan bilangan acak sebanyak 8 kali dengan :

|  |  |  |  |
| --- | --- | --- | --- |
| ꘘ t ꘘ | | ⁘t dan | 께 t |
| ⁘ | t 쑸 | ⁘t | 께 |
|  | 께 쑸 | ⁘t |  |
| t | 쑸 | ⁘t |  |
|  | 쑸 | ⁘t | ⁘께 |
|  | ⁘께 쑸 | ⁘t | ⁘ |
|  | ⁘ 쑸 | ⁘t |  |
|  | 쑸 | ⁘t | ⁘ |
|  | ⁘ 쑸 | ⁘t |  |

Bilangan acak yang dibangkitkan adalah : 께t t t ⁘t t ⁘ t

Dapat dilihat pada percobaan di atas tidak terdapat perulangan secara periodik.

### 2.5 *Android*

*Android* merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka seßndiri untuk digunanakan oleh bermacam piranti bergerak. *Android* umum digunakan pada *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS. *Android* tidak terikat ke satu merek *handphone* saja, beberapa vendor terkenal yang sudah memakai Android antara lain Samsung, XIAOMI, VIVO, OPPO, Motorolla, Lenovo dan lain-lain (Nazaruddin, 2012).

Pada Juli 2000, Google bekerjasama dengan Android *Inc.,* perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri *Android Inc.* bekerja pada Google, diantaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android *Inc*. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Versi *android* terbaru yaitu versi 9.0 (Pie)*.*

Berikut spesifikasi android versi 9.0 (Pie). *Android Pie* adalah pembaruan selanjutnya dari *Android* 8.1 "Oreo" yang merupakan pembaruan versi ke 9 dan pembaruan urutan ke-16 dari sistem operasi *Android*, untuk versi pengembangan nya pertama kali dirilis pada Maret 2018 dan dirilis ke publik pada 6 Agustus 2018.

### 2.6 *Unified Modeling Language* (UML)

[UML](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html) [(*Unified*](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html)[*Modeling*](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html)[*Language*)](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html) adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artefact* dapat berupa model, deskripsi atau perangkat lunak dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Shalahuddin, 2013)

#### 2.6.1 *Use Case Diagram*

*Diagram Use case* adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat *user.* Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. Simbol *Use Case Diagram* dapat dilihat pada Tabel 2.

**Tabel 2.2 Simbol *Use Case Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Actor* | Menspesifikasikan  himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use*  *case* |
| 2. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri  (*dependent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri  (independent) |

|  |  |  |  |
| --- | --- | --- | --- |
| 3. |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk  (*ancestor*) |
| 4. |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara  eksplisit |
| 5. |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan |
| 6. |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7. |  | *System* | Menspesifikasikan paket yang menampilkan sistem  secara terbatas |
| 8. |  | *Use case* | Deskripsi dari uraian aksiaksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor* |
| 9. |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen- |
|  |  |  | elemennya (sinergi) |
| 10. |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu  sumber daya komputasi |

#### 2.6.2 *Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). *Activity diagram* ini mirip dengan *flowchart diagram*. Simbol *Activity Diagram* dapat dilihat pada Tabel 3.

### Tabel 2.3 Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Activity* | Memperlihatkan bagaimana masingmasing kelas antarmuka saling berinteraksi satu sama lain. |
| 2. |  | *Action* | *State* dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3. |  | *Initial Node* | Bagaimana objek dibentuk atau diawali |
| 4. |  | *Activity*  *Final Node* | Bagaimana objek dibentuk dan diakhiri |
| 5. |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

#### 2.6.3 *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

**Tabel 2.4 Simbol S*equence Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-  informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-  informasi tentang aktifitas yang terjadi |
| 4 |  | *Message* | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy* |

#### 2.6.4 *Class Diagram*

*Class* adalah kumpulan objek-objek dengan dan yang mempunyai struktur umum, *behavior* umum, relasi umum, dan semantic/kata yang umum. *Class-class* ditentukan/ditemukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan.

**Tabel 2.5 Simbol *Class Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Association* | Apa yang menghubungkan antara objek satu dengan  objek lainnya |
| 2. |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek |
| 3. |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta  operasi yang sama |
| 4. |  | *Collaboration* | Deskripsi dari urutan aksiaksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu *actor* |
| 5. |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek |
| 6. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7. | 0..1 | *Nilai kardinitas* | Kosong atau satu. |
| 8. | 0..\* |  | Lebih dari sama dengan |
|  |  | *Nilai kardinitas* | kosong. |
| 9. | 0..n | *Nilai kardinitas* | Lebih dari sama dengan n, dimana n lebih dari satu. |
| 10. | 1 | *Nilai kardinitas* | Hanya satu |
| 11. | 1..\* | *Nilai kardinitas* | Lebih dari sama dengan satu |
| 12. | 1..n | *Nilai kardinitas* | Lebih dari sama dengan satu dimana n lebih dari satu. |
| 13. | \* | *Nilai kardinitas* | Banyak atau *Many.* |
| 14. | 1 | *Nilai kardinitas* | Hanya N, dimana N lebih dari satu. |
| 15. | n..\* | *Nilai kardinitas* | Lebih dari sama dengan N dimana N lebih dari satu. |
| 16. | n..m | *Nilai kardinitas* | Lebih dari sama dengan N dan kurang dari sama dengan M. Dimana M dan N lebih dari satu. |

#### 2.7 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. Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program/aplikasi (Oktaviani, 2018)

##### 2.7.1 *Typescript*

*Typescript* adalah bahasa pemrograman berbasis JavaScript yang menambahkan fitur *strong-typing* & konsep pemrograman OOP klasik (*class*, *interface*). Di dalam dokumentasinya, TypeScript disebut sebagai *super-set* dari JavaScript, artinya semua kode JavaScript adalah kode TypeScript juga. Bahasa pemrograman ini menawarkan *class*, *module*, dan *interface* yang membuat *developer* bisa mengembangkan aplikasi kompleks dengan lebih mudah. Hal inilah yang membedakannya dengan *javascript* (Hidayat, 2016)

TypeScript menawarkan class, module, dan interface yang membuat developer bisa mengembangkan aplikasi kompleks dengan lebih mudah diantara keunggulan dan fitur – fiturnya (Hidayat, 2016), antara lain :

1. *Support Class* dan *Module*
2. *Static Type-checking*
3. *Support* ES6 *Feature*
4. *Clear Library API Definition*
5. *Build-in Support* untuk JavaScript *Packaging*
6. Kesamaan *Syntax* untuk *Backend*
7. *Superset* dari JavaScript

##### 2.7.2 Angular-CLI

Angular-CLI merupakan generator untuk project Angular 2 dan 4 (serta versi diatasnya), dengan menggunakan bantuan Angular-CLI ini kita tidak perlu lagi *setup* banyak kebutuhan dasar seperti susunan *folder*, cara *run*, cara *build* dan lain-lain. Kesemuanya itu telah ada perintahnya (*command*) melalui Angular-CLI. Seperti halnya Ember.js yang mempunyai Ember-CLI, Vuejs dengan Vue-CLI dan React dengan “*create-react-app*” maka seperti itu pula fungsi Angular-CLI bagi Angular (Maulana, 2017).

#### 2.8 *Database*

Sistem *database* adalah suatu sistem penyusunan dan pengelolaan data dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan (Ahmad, 2007)

*Database* SQL (*Structure Query Language*) merupakan database relasional atau database yang terstruktur. Jadi, SQL database ini merupakan penyimpanan data yang menggunakan *relation* (hubungan) dan biasa disebut sebagai tabel. Artinya *Sql database ini* dapat menemukan data yang di carinya menggunakan karakteristik umum yang terdapat pada penyimpanannya dan hasil pencariannya disebut dengan skema (Kadir, 2015).

#### 2.9 SQLite

*Database* SQLite adalah solusi penyimpanan yang baik jika anda memiliki data terstruktur yang perlu diakses dan disimpan secara persisten serta sering ditelusuri dan diubah. Anda juga bisa menggunakan SQLite sebagai media penyimpanan utama untuk data aplikasi atau pengguna, atau anda juga bisa menggunakannya untuk proses caching serta menyediakan data yang diambil dari cloud.

Jika anda bisa menyatakan data berupa baris dan kolom, pertimbangkan untuk memakai database SQLite. Jika anda menggunakan database SQLite, yang dinyatakan sebagai objek SQLiteDatabase adalah semua interaksi dengan database adalah melalui instance dari kelas SQLiteOpenHelper yang akan mengeksekusi permintaan dan pengelolaan database (Ridwansyah, 2018).

#### 2.10 *Android SDK (Software Development Kid)*

Sesuai dengan namanya merupakan kumpulan tools-tools yang di perlukan untuk mengembangkan aplikasi Android. Seperti pada *java* yang dapat dijalankan pada *multiplatform*, SDK Android pun tersedia untuk beberapa platform seperti Windows, Mac OS dan Linux. Adapun aplikasi SDK yang didalamnya terdapat beberapa folder sebagai berikut (Mulyadi, 2010):

1. *Tools*
2. Berisikan tools-tools yang diperlukan dalam membangun aplikasi Android seperti emulator, *Android Virtual Device (AVD), SDK Manager, ddms* dan banyak lagi untuk memudahkan membangun aplikasi Android.
3. *Platforms*

Direktori ini ditujukkan untuk meletakkan versi platform-platform Android yang telah pengguna download dari *SDK Manager.* Seperti Android

1.5, 1.6, 2.0 dan seterusnya.

1. *Add-ons*

Direktori ini nantinya berisikan *tools* tambahan untuk Android SDK, yang fungsinya untuk menambahkan *library* yang akan digunakan untuk pengembangan aplikasi. Seperti jika ingin mengembangkan aplikasi menggunakan *map* dari google maka harus memerlukan platform Android yang sudah terdapat *library map.*

#### 2.11 Pengujian *Black Box*

Pengujian *black box* merupakan pendekatan komplementer dari teknik *white box*, karena pengujian *black box* diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik *white box*. Pengujian *black box* berfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program. Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Pengujian *black box* berusaha menemukan kesalahan dalam kategori (Rouf, 2012):

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface.*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Berbeda dengan pengujian *white box*, pengujian *black box* cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *black box* harus dapat menjawab pertanyaan sebagai berikut :

1. Bagaimana validasi fungsional diuji ?
2. Masukkan apa yang akan membuat kasus pengujian menjadi lebih baik ?
3. Apakah sistem akan sangat sensitif terhadap masukkan tertentu ?
4. Apakah aplikasi dapat bekerja disistem operasi yang berbeda ?

**BAB III**

**METODOLOGI PENELITIAN**

#### 3.1 Metode Pengumpulan Data

Beberapa metode yang digunakan dalam pengumpulan data adalah sebagai berikut:

1. Kajian Pustaka

Metode pengumpulan data yang digunakan yaitu kajian pustaka, digunakan untuk mencari *literature* atau sumber pustaka yang berkaitan dengan perangkat lunak yang dibuat dan membantu memperjelas teori-teori yang ada. Sumber literatur berupa buku, *e-book*, *paper*, jurnal, karya ilmiah, dan beberapa situs penunjang.

1. Wawancara (*Interview*)

Metode wawancara ini dilakukan dengan para guru PAUD, untuk mengumpulkan data dan informasi tentang hal-hal yang dibutuhkan dalam penelitian. Wawancara yang dilakukan lebih menitik beratkan pertanyaan mengenai ketertarikan anak terhadap pelajaran dasar dan hal yang dapat meningkatkan semangat belajar anak-anak.

#### 3.2 Metode Pengembangan Aplikasi

Dalam merancang atau membuat perangkat lunak dibutuhkan suatu pemodelan dari keseluruhan proses-proses yang akan dilakukan selama pembuatan perangkat lunak tersebut. Model yang digunakan dalam perangkat lunak ini adalah proses *Rational Unified Proses* (RUP). Adapun proses dari metode atau model RUP ini meliputi :

1. *Inception/permulaan*

Pada tahap ini dilakukan pemodelan ruang lingkup proyek yang akan dibuat.

Adapun ruang lingkup dari proyek ini adalah sebagai berikut:

1. *Game* edukasi ini terdiri dari kuis :

* 1. Pengenalan huruf dengan model permainan penyusunan huruf menjadi kata.
  2. Penjumlahan angka dengan model permainan penyusunan angka untuk mencapai jumlah yang telah ditentukan.
  3. Pengenalan gambar hewan melalui suara hewan yang diperdengarkan.
  4. Pengenalan suara hewan berdasarkan gambar hewan yang ditampilkan.
  5. Pengenalan warna dengan menampilkan benda yang memiliki ciri khas warnanya.

1. Setiap kuis akan memiliki waktu pengerjaan dan tingkat kesulitan yang berbeda.
2. Setiap kuis akan menampilkan hasil dari jawaban yang telah dimasukkan dan akan menampilkan jawaban yang benar.
3. Dalam pembuatannya menggunakan bahasa pemrograman *Typescript* dan penyimpanan data menggunakan *SQLite*.
4. Algoritma yang digunakan adalah *Linear Congruent Method* (LCM) untuk pengacakan soal.
5. Aplikasi ini berjalan disistem operasi *android* versi 5.0 ke atas.
6. Aplikasi berjalan secara *offline.*
7. *Elaboration/elaborasi*

Pada tahap ini, setelah menentukan ruang lingkup proyek, selanjutnya akan dilakukan langkah-langkah berikut :

* 1. Melakukan analisis masalah menggambarkan alur sistem yang akan dibuat/diusulkan.
  2. Membuat perancangan aplikasi menggunakan UML (*Unified Modelling Language*.
  3. Membuat perancangan tampilan (*interface*) untuk setiap tampilan

*form*/halaman yang ada dalam sistem yang akan dibuat.

1. *Construction/konstruksi*

Pada tahap ini, penulis melakukan tahap implementasi yaitu sebagai berikut: a. Melakukan pengkodean (*coding*) baik sisi *front end* dan *back end.*

b. Melakukan testing hasil pengacakan dari metode *Linear Congruent*

*Method*

1. *Transition/transisi*

Menguji aplikasi yang telah dibuat dan memperbaiki masalah-masalah yang muncul saat dan setelah pengujian.

#### 3.3 Waktu dan Tempat Penelitian

##### 3.3.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Juni 2020. Rincian kegiatan dapat dilihat pada tabel dibawah ini : **Tabel 3.1 *Gantt Chart* Waktu Penelitian**

No

Uraian

Waktu

Juli

Agustus

September

Oktober

November

III

IV

I

II

III

IV

I

II

III

IV

I

II

III

IV

I

II

1

*Inception*

2

*Elaboration*

3

*Construction*

4

*Transition*

##### 3.3.2 Tempat Penelitian

Penelitian tugas akhir bertempat di Taman Kanak – Kanak Pembina 1 Jl.

Sao Sao, Kecamatan Bende Kota Kendari.

#### 3.4 Analisis Sistem

Analisis sistem merupakan suatu tahapan yang bertujuan untuk mengetahui dan mengamati apa saja yang terlibat dalam suatu sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis masalah, analisis kebutuhan fungsional dan analisis kebutuhan nonfungsional.

Dalam tugas akhir ini, berdasarkan studi literatur dan hasil observasi yang dilakukan, dari hasil pengamatan diperoleh permasalahan yaitu bagaimana membuat *game* edukasi yang dapat meningkatkan ketrampilan dan kemampuan anak dalam mengenal huruf, angka, gambar, suara dan warna pada anak usia dini. *Game* edukasi berbasis android dikarenakan *smartphone* android paling banyak dan mudah digunakan oleh anak-anak. Untuk mengatasi kejenuhan dalam bermain dan agar soal tidak monoton perlu digunakannya metode pengacakan yaitu *Linear Congruent Method (LCM)* untuk pengacakan soal kuis.

##### 3.4.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun. Setelah melalui tahapan analisis, maka telah ditetapkan kebutuhan-kebutuhan untuk membangun sistem meliputi *input*, proses dan *output.*

###### 3.4.1.1 Analisis Kebutuhan *Input*

*Input* dari aplikasi yang di bangun terdiri dari:

1. Soal kuis yang berupa data huruf, angka, gambar, dan suara
2. Pilihan jawaban dan pembahasan dari soal.

###### 3.4.1.2 Analisis Kebutuhan Proses

Kebutuhan proses dalam aplikasi ini antara lain:

1. Proses pembuatan aplikasi menggunakan Bahasa pemrograman *Typescript* dan *database SQLite.*
2. Proses pengacakan soal menggunakan algoritma *Linear Congruent Method (LCM).*

###### 3.4.1.3 Analisis Kebutuhan *Output*

*Output* yang diperoleh dalam sistem ini adalah sebuah *game* edukasi untuk pengenalan huruf, angka, gambar,suara dan warna berbasis android untuk anak usia dini dengan metode *Linear Congruent Method (LCM)* untuk pengacakan soal.

##### 3.4.2 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan nonfungsional adalah sebuah langkah dimana seorang pembangun aplikasi menganalisis sumber daya yang dibutuhkan untuk membangun aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

###### 3.4.2.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan pada pembangunan sistem ini, sebagai berikut:

**Tabel 4.1 Spesifikasi Perangkat Keras**

|  |  |  |
| --- | --- | --- |
| No. | Nama Perangkat | Spesifikasi |
| 1. | *Notebook* | *Asus* |
| 2. | *Processor* | *Intel Core i5 @1.80 GHz* |
| 3. | *Monitor* | *Monitor 14 inch* |
| 4. | *Memori* | *RAM 4 GB DDR 3* |
| 5. | *Harddisk* | *500 GB HDD* |
| 6. | *Smartphone* | *Android 5.0, RAM 2 GB* |

###### 3.4.2.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan pada pembangunan ini, sebagai berikut:

**Tabel 4.2 Spesifikasi Perangkat Lunak**

|  |  |  |
| --- | --- | --- |
| No. | Nama Perangkat | Deskripsi |
| 1. | *Operating System Windows 10* | *Windows* 8 digunakan untuk menjalankan aplikasiaplikasi yang dibutuhkan dalam pembuatan program |
| 2. | *Android SDK* | *Android SDK* versi 5.0 digunakan untuk menjalankan aplikasi android yang dibuat |
| 3. | *SQLite* | Sistem manajemen *database* relasional, tanpa memerlukan server yang terpisah dari aplikasi |
| 4. | *Mozila Firefox* | Browser untuk mengakses antarmuka dari SQLite |
| 5. | *Visual Studio* | Text editor untuk mengkoding program |

#### 3.5 *Unified Modeling Language* (UML)

Aplikasi dibangun dengan menggunakan *Unified Modeling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram, Activty Diagram, Sequence Diagram* dan *Class Diagram*.

##### 3.5 1 Perancangan *Use Case Diagram*

*Use Case diagram* menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. Berikut ini adalah *use case diagram* untuk *game* edukasi.



# Gambar 4.1 *Use Case Diagram*

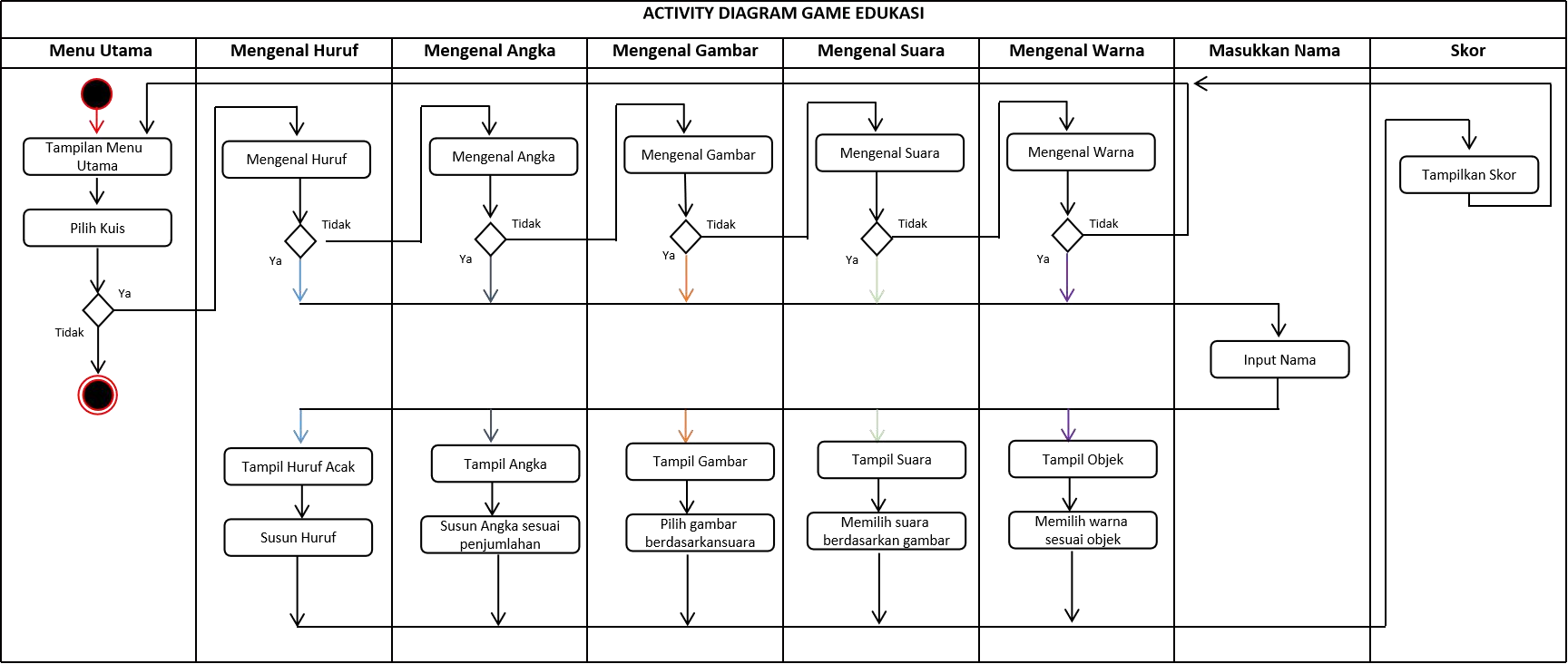
**Tabel 4.3 Definisi *Use Case Diagram***

|  |  |  |
| --- | --- | --- |
| **No** | ***Use Case*** | **Deskripsi** |
| 1. | Memilih kuis huruf | *Use case* ini digunakan untuk memilih kuis huruf |
| 2. | Memilih kuis angka | *Use case* ini digunakan untuk memilih kuis angka |
| 3. | Memilih kuis gambar | *Use case* ini digunakan untuk memilih kuis gambar |
| 4. | Memilih kuis suara | *Use case* ini digunakan untuk memilih uis suara |
| 5. | Memilih kuis warna | *Use case* ini digunakan untuk memilih kuis warna |
| 6. | Masukkan Nama | *Use case* ini digunakan untuk merekam inputan nama pemain/*user.* |
| 7. | Skor | *Use case* ini digunakan untuk menampilkan skor atau nilai dari setiap kuis dan setiap pemain *user* |
| 8. | *About Apps* | *Use case* ini digunakan untuk menampilkan informasi mengenai aplikasi |

#### 3.5.2. Perancangan *Activity Diagram*

a. *Activity Diagram Game Edukasi*

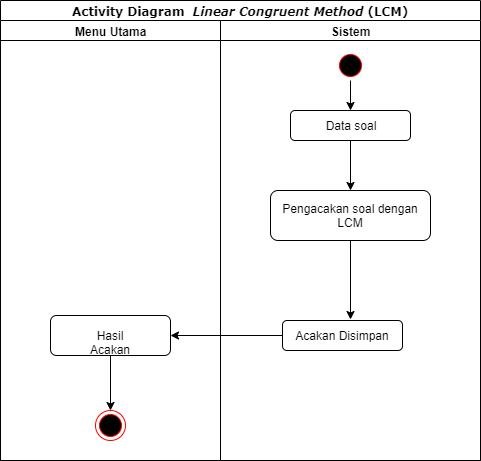
Pada Gambar 4.3 menunjukkan *Activity Diagram Game* edukasi, alur memilih kuis dan melihat skor tiap kuis. *User* dapat memilih kuis dan memainkannya, dihasil akhir akan terlihat nilai atau skor masing-masing kuis.



# Gambar 4.2 *Activity Diagram Game* Edukasi

*b. Activity Diagram LCM (Linear Congruent Method)*

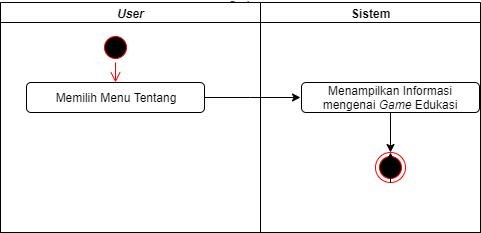
Pada Gambar 4.4 menunjukkan *Activity Diagram* penerapan *Linear Congruent Method* yang digunakan.



# Gambar 4.3 *Activity Diagram Linear Congruent Method*

*c. Activity Diagram* Tentang

Pada Gambar 4.5 menunjukkan *Activity Diagram* menu tentang, *user* dapat melihat informasi mengenai aplikasi dan pembuat aplikasi.



# Gambar 4.4 *Activity Diagram Game* Edukasi

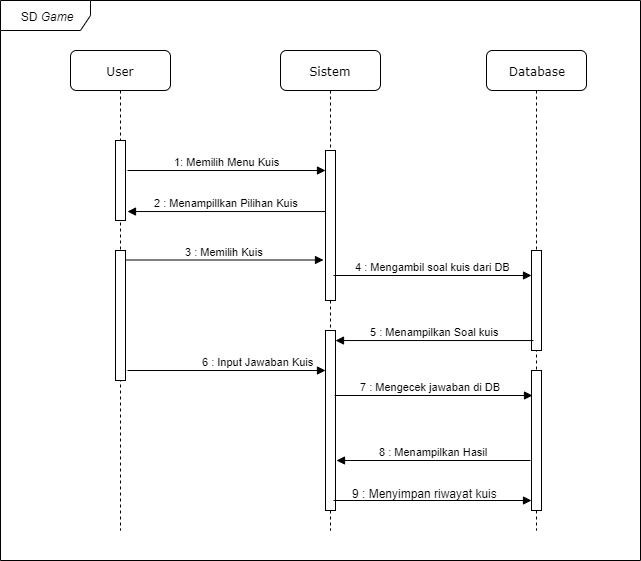
#### 3.5.3 Perancangan *Sequence Diagram*

*a. Sequence Diagram* Tentang



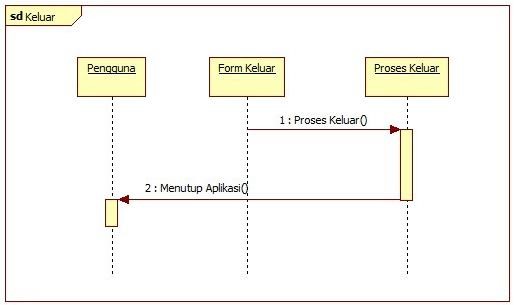
# Gambar 4.5 *Sequence Diagram* Tentang

*b. Sequence Game* Edukasi



# Gambar 4.6 *Sequence Game Diagram* Edukasi

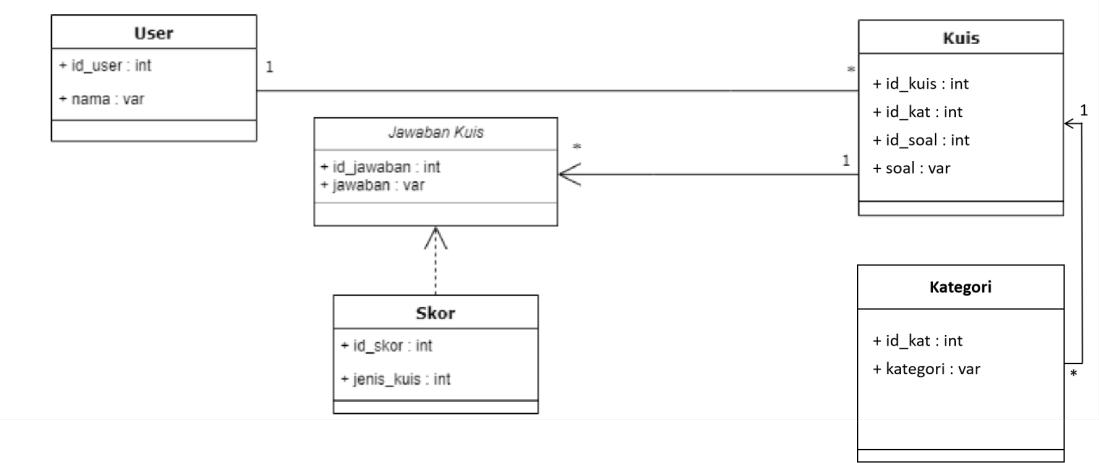
*c. Sequence Diagram* Keluar



# Gambar 4.7 *Sequence Diagram* Keluar

#### 3.5.4 Perancangan *Class Diagram*

Pada Gambar 4.11 menunjukkan *Class Diagram Game Edukasi,* meliputi tabel *user,* table kuis, tabel jawaban kuis, tabel skor.



# Gambar 4.8 *Class Diagram Game* Edukasi

#### 3.6 Metode Pengujian

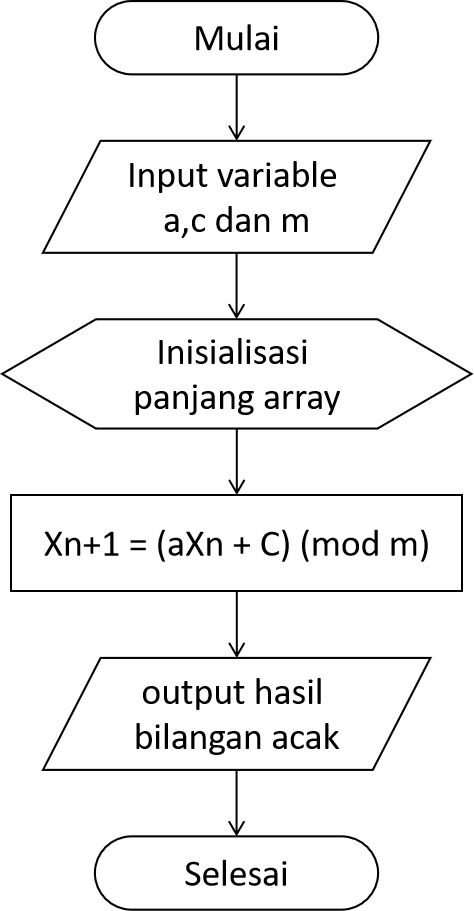
Untuk pengujian aplikasi media pembelajaran berbasis mobile digunakan metode black box testing. Black Box Testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan cek fungsional perangkat lunak. Tujuan Black Box Testing adalah untuk mencari kesalahan/kegagalan dalam operasi tingkat tinggi, yang mencakup kemampuan dari perangkat lunak, operasional/tata laksana, skenario pemakai. Fungsi dari pengujian ini berdasarkan kepada apa yang dapat dilakukan oleh sistem.

**Tabel 4.4. Rancangan Pengujian Aplikasi**

|  |  |  |
| --- | --- | --- |
| **Kelas Uji** | **Detail Uji** | **Jenis Pengujian** |
| **Pengujian Fitur** | Input Nama User | *Black Box* |
|  | Tampilkan Skor | *Black Box* |
|  | Menjawab Soal | *Black Box* |
|  | Logout | *Black Box* |
| **Pengujian Menu** | Kategori Huruf | *Black Box* |
|  | Kategori Angka | *Black Box* |
|  | Kategori Gambar | *Black Box* |
|  | Kategori Suara | *Black Box* |
|  | Kategori Warna | *Black Box* |
| **Pengujian Algoritma** | Pengacakan Soal | *White Box & Black Box* |

#### 3.7 Perancangan Algoritma *Linear Congruent Method*

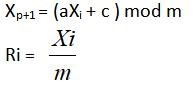
Pada gambar 4.12 dijelaskan langkah pertama dalam *linear congruent method* adalah melakukan input variabel a, c, dan m, pemilihan konstanta a, c, dan m menentukan hasil bilangan acak yang benar-benar tidak terjadi pengulangan. Setelah menentukan konstanta a, c dan m dilanjutkan menentukan panjang bilangan acaknya dan langsung bilangan tersebut dimasukan ke rumus dari algoritma LCM maka akan kelur hasil bilangan acak yang tidak terjadi pengulangan.



# Gambar 4.9 *Flowchart* Algoritma *Linear Congruent Method*

## Perhitungan Manual

Dalam metode *Linear Congruent Method* terdapat 4 variabel yang di digunakan untuk membangkitkan bilang acak, implementasi variabel metode *Linear Congruent Method* pada sistem adalah dengan menentukan secara statis variabel a, c dan m sedangkan elemen *seed /* Xo menggunakan fungsi random bawaan pada *javascript* dan tidak merupakan keliapatan 5 sehingga elemen s*eed* yang diberikan oleh sistem akan menghasilkan angka – angka yang nanti nya akan digunakan sebagai penentuan urutan munculnya soal dari permainan.

....................................... **(1)**

Dimana : Xo = Elemen Seed

a = Pengali c = Penambah m = modulus c = 0 => multiplicative LCM

c ≠ 0 => mixed LCM

Kunci pembangkit adalah Xo atau elemen *seed* yang merupakan nilai pertama kali yang muncul pada pengacakan yang di dapat secara acak. LCM mempunyai periode tidak lebih besar dari m, dan pada kebanyakan kasus periodenya kurang dari itu.

LCM mempunyai periode penuh ( − 1) jika memenuhi syarat berikut:

1. relative prima terhadap .
2. – 1 dapat dibagi dengan semua faktor prima dari
3. – 1 adalah kelipatan 4 jika m adalah kelipatan 4
4. > ( , , 0) 5. > 0, > 0

Meskipun LCM secara teoritis mampu menghasilkan bilangan acak yang lumayan, namun sangat sensitif terhadap pemilihan nilai-nilai a, b, dan m. pemilihan nilainilai yang tidak sesuai dapat mempengaruhi implementasi pada LCM.

### 3.6.1 Kasus 1 (X = 27, a = 17, c = 43, m = 100)

X1 = (17.27 + 43) mod 100

= 502 mod 100

X1 = 2

X2 = (17.2+ 43) mod 100

= 77 mod 100

X2 = 77

X3 = (17.77 + 43) mod 100

= 1352 mod 100

X3 = 52

X4 = (17.52 + 43) mod 100

= 927 mod 100

X4 = 27

X5 = (17.27 + 43) mod 100

= 502 mod 100

X5 = 2

X6 = (17.2+ 43) mod 100

= 77 mod 100

X6 = 77

X7 = (17.77 + 43) mod 100

= 1352 mod 100

X7 = 52

X8 = (17.52 + 43) mod 100

= 927 mod 100 X8 = 27

Dari konfigurasi diatas ditemukan urutan data 2, 77, 52, 27, 2, 77, 52, 27 dengan konfigurasi diatas hanya dapat memunculkan 4 angka yang teracak danhasil selanjutnya yang terus berulang.

### 3.6.2 Kasus 2 (X = 4, a = 2, c = 10, m = 20)

X1 = (2.4 + 10) mod 20 X1 = 18

X2 = (2.18+ 10) mod 20 X2 = 6

X3 = (2.6+ 10) mod 20 X3 = 2

X4 = (2.2+ 10) mod 20 X4 = 14

X5 = (2.14+ 10) mod 20 X5 = 18

X6 = (2.18+ 10) mod 20 X6 = 6

X7 = (2.6+ 10) mod 20 X7 = 2

X8 = (2.2+ 10) mod 20 X8 = 14

X9 = (2.14+ 10) mod 20 X9 = 18

X10 = (2.18+ 10) mod 20

X10 = 6

X11 = (2.6+ 10) mod 20

X11 = 2

Dari konfigurasi diatas ditemukan urutan data 18, 6, 2, 14, 18, 6, 2, 14, 18, 6, 2 dengan konfigurasi diatas hanya dapat memunculkan 4 angka yang teracak dan hasil selanjutnya yang terus berulang.

### 4.4.3 Kasus 3 (X = 7, a = 3, c = 5, m = 25)

X1 = (3.7 + 5) mod 25 X1 = 1

X2 = (3.1 + 5) mod 25 X2 = 8

X3 = (3.8 + 5) mod 25 X3 = 4

X4 = (3.4 + 5) mod 25 X4 = 17

X5 = (3.17 + 5) mod 25 X5 = 6

X6 = (3.6 + 5) mod 25 X6 = 23

X7 = (3.23 + 5) mod 25 X7 = 24

X8 = (3.24 + 5) mod 25 X8 = 2

X9 = (3.2 + 5) mod 25 X9 = 11

X10 = (3.11 + 5) mod 25 X10 = 13

X11 = (2.14 + 5) mod 25

X11 = 19

Dari konfigurasi diatas ditemukan urutan data 1, 8, 4, 17, 6, 23, 24, 2, 11, 13, 19 dengan konfigurasi diatas hanya dapat memunculkan 11 angka yang teracak dan belum terdapat pengulangan dengan mengikuti 4 syarat diatas.

### 4.5 Perancangan Sistem

Perancangan sistem merupakan suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Tahapan perancangan sistem membuat tampilan *mockup game* edukasi yang akan dibangun untuk menjadi acuan dalam proses pengembangan. Rancangan *user interface* ini dibuat sederhana agar mudah dimengerti oleh *user* dan tidak ada kerumitan dalam menjalankannya sehingga mencapai tujuan perangkat lunak yang *user friendly.*

1. Halaman *Dashboard*

Pada halaman *dashboard* menampilkan pilihan kuis yang dapat dipilih yaitu huruf, angka, gambar, suara, dan warna. Pada halaman *dashboard* juga menampilkan menu skor untuk melihat skor dari setiap kuis.



# Gambar 4.10 Tampilan *Dashboard*

2. Halaman Mulai *Game*

Pada halaman ini *user* akan diminta memasukkan nama untuk dicatat dalam papan skor. Kemudian *user* atau pemain menekan tombol mulai.



# Gambar 4.11 Tampilan Mulai Game

3. Halaman Kuis Huruf

Pada halaman kuis huruf, akan menampilkan huruf teracak dan pertanyaan untuk menyusun huruf-hurf yang teracak menjadi sebuah kata yang merupakan jawaban dari pertanyaan yang diberikan.



# Gambar 4.12 Tampilan Kuis Huruf

4. Halaman Skor

Pada halaman skor menampilkan skor atau nilai tiap pemain berdasarkan tiap kuis yang telah dimainkan.



## Gambar 3.17 Tampilan Skor Tiap Kuis

### 3.7 Perancangan Soal Quiz

Perancangan soal pada sistem dilakukan dengan berkonsultasi pada Dosen Ahli Pendidikan Anak Usia Dini sehingga didapatkan total 250 soal yang terbagi menjadi 5 kategori dan 2 tingkat kesusahan. Pada tingkat soal mudah ditargetkan untuk anak usia 3 - 4 tahun dan pada tingka soal susah ditargetkan 5 - 6 tahun.

Berikut tabel pembagian jumlah soal berdasarkan kategori dan tingkat kesusahan

**Tabel 4.5. Soal Quiz**

|  |  |  |
| --- | --- | --- |
| **Kategori** | **Tingkat Kesusahan** | **Jumlah Soal** |
| **Kategori Gambar** | Mudah (3 - 4 Tahun) | 25 Soal |
| Susah (5 - 6 Tahun) | 25 Soal |
| **Kategori Huruf** | Mudah (3 - 4 Tahun) | 25 Soal |
| Susah (5 - 6 Tahun) | 25 Soal |
| **Kategori Angka** | Mudah (3 - 4 Tahun) | 25 Soal |
| Susah (5 - 6 Tahun) | 25 Soal |
| **Kategori Suara** | Mudah (3 - 4 Tahun) | 25 Soal |
| Susah (5 - 6 Tahun) | 25 Soal |
| **Kategori Warna** | Mudah (3 - 4 Tahun) | 25 Soal |
| Susah (5 - 6 Tahun) | 25 Soal |

**DAFTAR PUSTAKA**

Andreas Parry, L. (2017). Analisis & Perbandingan Blum Blum Shub dan Inversive Congruential Generator Beserta Implementasinya. Jurusan Teknik Informatika ITB, Bandung

Andresta, R. (2017). Perbandingan Algoritma Linear Congruential Generators

BlumBlumShub, dan MersenneTwister untuk Membangkitkan Bilangan Acak Semu. Rajawali Press. Jakarta

Arizqia, M.G. and Widodo, A.A., 2017. Rancang Bangun Aplikasi Dengan Linear Congruent Method (LCM) Sebagai Pengacakan Soal. *JOINTECS (Journal of Information Technology and Computer Science)*, *2*(1).

Fishman, G.S. and Moore, III, L.R., 1986. An Exhaustive Analysis of Multiplicative

Congruential Random Number Generators with Modulus 2^31-1. *SIAM*

*Journal on Scientific and Statistical Computing*, *7*(1), pp.24-45.

Gentle, J.E., 2003. Monte carlo methods. *Random number generation and Monte Carlo methods*, pp.229-281.

Hartati, S., 2005. Konsep Dasar Pendidikan Anak Usia Dini. *Jakarta: Universitas Negeri Jakarta*.

Habibi, M.M., 2018. *Analisis kebutuhan anak usia dini (buku ajar S1 PAUD)*.

Deepublish.

Hurd, D. and Jenuings, E., 2009. Perancangan Education Game Yang Baik.

Ikhsan, M. and Dahria, M., Sulindawaty (2011). *Penerapan Association Rules Dengan Algoritma Apriori Pada Proses Pengelompokan Barang di*

*Perusahaan Retail*.

Kakiay, T.J., 2004. Dasar Teori Antrian Untuk Kehidupan Nyata. *Yogyakarta: Andi*.

Mansur, M.A., 2005. Pendidikan anak usia dini dalam Islam. *Yogyakarta: Pustaka*

*Pelajar*.

Mulyasa, H.E., 2012. Manajemen Paud. *Bandung: Remaja Rosdakarya*.

Pendidikan, K.K., 2013. Kementerian Pendidikan dan Kebudayaan.

Putra, D.W., Nugroho, A.P. and Puspitarini, E.W., 2016. Game Edukasi berbasis android sebagai media pembelajaran untuk anak usia dini. *JIMP-Jurnal Informatika Merdeka Pasuruan*, *1*(1).

89

Sagala, S., 2009. *Kemampuan Profesional Guru dan Tenaga Kependidikan: Pemberdayaan guru, tenaga kependidikan, dan masyarakat dalam manajemen sekolah*. Alfabeta.

Sanjaya, W. (2008). Kurikulum dan Pembelajaran:Teori dan Praktik Pengembangan Kurikulum Tingkat Satuan Pendidikan. Kencana Prenada Media Group.

Shalahudin, S., 2013. Pemodelan Visual dengan UML. *Yogyakarta: Graha Ilmu*.

Siegel, Sidney, 1986, Statistik Non Parametrik untuk Ilmu - ilmu Sosial, Terjemahan, Jakarta : PT.Gramedia

Yandi, 2014, Pembangunan Game Edukasi Sistem Pencernaan Manusia Berbasis Dekstop, Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia, Bandung

Yani, E.C., Boko, S. and Funny, F.C., 2016. *Rancang Bangun Aplikasi Edugame Slider Puzzle Keanekaragaman Budaya Bengkulu Dengan Menggunakan Linear Congruent Method (LCM)* (Doctoral dissertation, Universitas Bengkulu).

Zapata, B.C., 2013. *Android studio application development*. Packt Publishing Ltd, New York.