

IMPLEMENTASI *ALGORITMA FISHER YATES* PADA *GAMES* EDUKASI PENGENALAN HEWAN UNTUK ANAK SD BERBASIS MOBILE ANDROID

Rio Andriyat Krisdiawan¹, Tri Ramdhany²

¹). Teknik Informatika Universitas Kuningan. ²). Sistem Informasi STMIK LPKIA Bandung
¹). Jln.Cut Nyak Dhien No.36 A Cijoho Kab.Kuningan, ²). Jl. Soekarno Hatta No.456 Bandung
¹). Email : rioandriyat@gmail.com, ²). triramdhany@gmail.com

Abstrak

Perkembangan teknologi yang semakin berkembang saat ini sangat berperan penting dalam perkembangan penggunaan perangkat mobile. Salah satu pemanfaatan teknologi mobile belakangan ini terutama anak-anak lebih memanfaatkannya dalam permainan atau game. Dengan maraknya perkembangan game yang semakin pesat, banyak game yang lebih focus kepada unsur hiburan dan kesenangan semata. Masih sedikit game yang memberikan edukasi yang bermanfaat bagi perkembangan edukasi anak. Dalam hal tersebut, penulis berinisiatif untuk membuat sebuah aplikasi *game* edukasi yang mampu memberikan manfaat dan memberikan nilai edukatif didalamnya yang baik untuk menambah pengetahuan anak. Dalam perancangan *game puzzle* membutuhkan suatu algoritma pengacakan yang dapat memberikan hasil yang baik didalam pengacakannya. Algoritma yang digunakan dalam penelitian ini yaitu algoritma *fisher yates*. Yaitu sebuah algoritma yang memanfaatkan fungsi *random*, sehingga nilai yang dihasilkan oleh algoritma *fisher yates* ini mampu memberikan hasil yang bias (hasil pengacakan oleh *fisher yates* sulit untuk diprediksi pola keluarannya). Untuk pembangunan aplikasi ini menggunakan metode pengembangan system RUP (*Rational Unified Process*). Dengan perancangan berbasis objek menggunakan UML (*Unified Modelling Language*) dan Bahasa pemrograman yang akan dibunakan adalah Bahasa pemrograman berbasis mobile android.

Kata Kunci : *Games, Algoritma, Fisher Yates, Mobile Android, Games Puzzle, Games Edukasi.*

1. PENDAHULUAN

Perkembangan teknologi yang semakin berkembang saat ini sangat berperan penting dalam perkembangan penggunaan perangkat mobile. Penggunaan perangkat mobile sudah menjadi kebutuhan yang sangat penting bagi manusia. Salah satu pemanfaatan teknologi mobile belakangan ini adalah berupa game, oleh karena itu penulis berinisiatif untuk membuat sebuah aplikasi *game* edukasi yang mampu memberikan manfaat dan memberikan nilai edukatif didalamnya yang baik untuk menambah pengetahuan anak. Dalam hal ini game yang akan dibuat dirancang dalam bentuk *puzzle*. *Puzzle* yaitu sebuah *game* yang menampilkan potongan – potongan gambar dimana potongan gambar tersebut teracak susunannya sehingga memberikan tantangan tersendiri untuk menyusunnya. Dalam perancangan *game puzzle* membutuhkan suatu algoritma pengacakan yang dapat memberikan hasil yang baik didalam pengacakannya. Selain itu algoritma pengacakan dipilih untuk memberikan tantangan yang berbeda di setiap *stage* yang dimainkan. Algoritma yang digunakan dalam penelitian ini yaitu algoritma *fisher yates*. Yaitu sebuah algoritma yang memanfaatkan fungsi *random*, sehingga nilai yang dihasilkan oleh algoritma *fisher yates* ini mampu memberikan hasil yang bias (hasil pengacakan oleh *fisher yates* sulit untuk diprediksi pola keluarannya). Untuk pembangunan aplikasi ini menggunakan metode pengembangan system RUP (*Rational Unified Process*). Dengan perancangan berbasis

objek menggunakan UML (*Unified Modelling Language*) dan Bahasa pemrograman yang akan dibunakan adalah Bahasa pemrograman berbasis mobile android.

Dari latar belakang diatas, maka dapat diketahui identifikasi masalah sebagai berikut :

1. Bagaimana mengimplementasikan algoritma
 1. *Fisher Yates* dalam sebuah game edukasi yang lebih menarik sehingga menambah minat pengguna untuk memainkannya.
 2. Bagaimana cara kerja algoritma *Fisher Yates* dalam mengacak susunan *puzzle games*.
2. Batasan masalah dalam penelitian ini adalah:
 1. Jenis game *puzzle* pada aplikasi *game* edukasi ini adalah *puzzle slide*.
 2. Aplikasi hanya di fokuskan pada unsur permainan dan edukasi yang ada didalamnya.
 3. Potongan-potongan gambar yang digunakan dalam *puzzle* berupa gambar hewan sehingga edukasi yang terkandung di dalam aplikasi *game* edukasi ini hanya seputar informasi hewan yang berguna untuk anak usia 6 tahun.
 4. Potongan *puzzle* akan menggunakan ukuran 2 x 2 untuk level yang paling mudah, 3x3 untuk level sedang dan potongan ukuran 4 x 4 untuk level yang paling sulit.
 5. Aplikasi *game* edukasi ini akan dibangun dengan memanfaatkan algoritma *fisher yates* sebagai pengacakan untuk urutan potongan gambar *puzzle*.

6. Penilaian pada *game puzzle* ini hanya menggunakan 3 bintang untuk setiap gambar *puzzle*.
7. Aplikasi *game* edukasi berbentuk *puzzle* ini dibangun menggunakan bahasa pemrograman Java berbasis Android.

Maksud dan Tujuan

Maksud dari penelitian ini adalah untuk membuat aplikasi *game* yang bermanfaat untuk edukasi anak sehingga mampu memberikan pengetahuan tentang mengenal jenis hewan pada anak usia 6 tahun.

Tujuan dari penelitian ini adalah untuk merancang dan membangun sebuah aplikasi *game* edukasi berbentuk *puzzle* dengan menerapkan algoritma *fisher yates* yang dapat menarik minat anak untuk bermain dan belajar yang berguna bagi pengetahuannya.

2. DASAR TEORI

Definisi Game

Game atau permainan adalah bagian mutlak dari kehidupan anak dan permainan merupakan bagian integral dari proses pembentukan kepribadian anak. *Game* dikenal sebagai sarana hiburan yang baik untuk mengembangkan daya kreatifitas pemainnya. Aplikasi permainan atau *game application* merupakan salah satu implementasi dari ilmu dibidang komputer. Inti dari sebuah aplikasi permainan adalah mengembangkan kemampuan otak untuk mengatur strategi, kecepatan, dan ketepatan dalam mencapai tujuan akhir.

Game Puzzle

Puzzle slide adalah representasi permainan yang dapat diselesaikan dengan mengurutkan atau menyusun komponen-komponen pembentuknya dengan kondisi yang berurut (Haryanto, 2010). *Puzzle* terdiri dari kotak-kotak angka yang dapat diacak sedemikian rupa sehingga menyusun pola *random* yang dapat dicari permasalahannya. Aplikasi permainan yang baik adalah aplikasi permainan yang dapat memberikan kreatifitas yang baik bagi penggunaanya,serta mampu memberikan pembelajaran yang tepat bagi penggunaanya, aplikasi permainan dapat memberikan pengaruh yang buruk karena dengan bermain terkadang penggunaanya terlelah dengan keadaan sekitar dan lupa akan dunia yang sebenarnya. permainan edukasi atau *game education* merupakan permainan yang telah dirancang khusus untuk mengajar orang tentang suatu subjek tertentu, memperluas konsep, memperkuat pembangunan, memahami sebuah peristiwa historis atau budaya, atau membantu mereka dalam mempelajari keterampilan dalam bermain. Aplikasi

permainan yang dikemas dalam konteks pendidikan ini diharapkan mampu memberikan semangat khususnya untuk anak-anak didalam belajar dengan menghiraukan rasa jenuh dengan materi yang sedang dipelajari.

Algoritma

Algoritma berasal dari nama seorang pengarang berkebangsaan Arab yang bernama Abu Ja'far Mohammed ibnu Musa al Khowarizmi (790 – 840) yang terkenal dengan sebutan bapak Aljabar. Algoritma adalah langkah detail yang ditunjukkan untuk komputer guna untuk menyelesaikan suatu masalah (Kadir A, 2012). Didalam proses analisis masalah diperlukan tindakan untuk mengidentifikasi informasin yang menjadi keluaran pemecahan masalah untuk mengidentifikasi informasi yang menjadi keluaran pemecahan masalah dan data-data yang menjadi masukan karena itu diperlukan prosedur yang mengolah masukan tersebut menjadi keluaran yang dikehendaki. Karena computer tidak akan berarti apa-apa tanpa adanya langkah detail yang menyusun prosedur (Kadir A, 2012).

Algoritma Fisher Yates

Fisher-Yates shuffle (diambil dari nama Ronald Fisher dan Frank Yates) atau juga dikenal dengan nama Knuth shuffle (diambil dari nama Donald Knuth), adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Fisher-Yates shuffle jika diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan berat sebelah, sehingga setiap permutasi memiliki kemungkinan yang sama (Ade-Ibijola, 2012). Algoritma ini menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tertentu. Metode dasar yang digunakan untuk menghasilkan permutasi acak untuk angka 1 sampai N (Ade-Ibijola, 2012). adalah sebagai berikut :

1. Tuliskan angka dari 1 sampai N
2. Pilih sebuah angka acak A diantara 1 sampai dengan angka N.
3. Dihitung dari bawah, tukarkan angka A dengan angka N terakhir, dan tuliskan angka tersebut dilain tempat.
4. Ulangi langkah 2 dan langkah 3 sampai semua angka sudah ditukarkan
5. Urutkan angka yang dituliskan pada langkah 3 adalah permutasi acak dari angka awal

Berikut adalah contoh pengerjaan algoritma fisher yates, sebagai keterangan Range adalah jumlah angka yang belum terpilih, roll adalah angka acak yang

terpilih, scratch adalah daftar angka yang belum terpilih dan result adalah hasil permutasi yang didapatkan.

Tabel 1. Contoh pengerjaan *Algoritma Fisher Yates*

Range	Roll	Scratch	Result
		12345678	
1-8	5	1234867	5
1-7	4	123786	45
1-6	1	62378	145
1-5	6	8237	6145
1-4	3	827	36145
1-3	7	82	736145
1-2	8	2	8736145

Permutasi yang didapatkan pada table contoh pengerjaan algoritma fisher yates adalah 28736145.

Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah keluarga notasi grafis yang didukung oleh model-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. (Martin Fowler, 2004).

Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena *UML* juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, *UML* tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Metode Pengembangan Sistem

Metodologi penelitian yang digunakan yakni tahapan-tahapan yang akan dilakukan dalam proses penyelesaian penelitian dalam rangka untuk memudahkan memecahkan masalah dari awal perencanaan strategis hingga tercapainya tujuan.

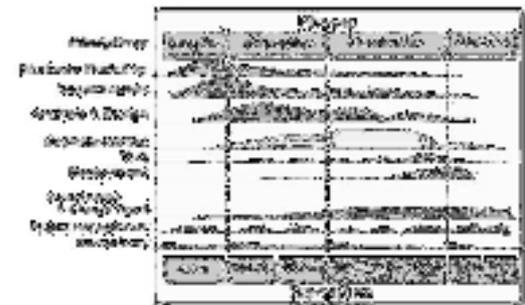
Metodologi yang digunakan oleh penulis dalam melakukan penelitian ini adalah *RUP (Rational Unified Process)*. *Rational Unified Process* merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak.

Model RUP sangat bagus digunakan untuk proses pengembangan perangkat lunak berbasis *Unified*

Modeling Language (UML). Hal ini dikarenakan metode RUP memakai cara-cara *Object Oriented Programming (OOP)* dalam membagi tahapan demi tahapan dan iterasi antar komponen yang terlibat.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language (UML)*. Melalui gambar 3.1 dapat dilihat bahwa RUP memiliki 2 dimensi pengembangan, yaitu:

1. **Dimensi pertama** digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception, Elaboration, Construction, dan Transition*.
2. **Dimensi kedua** digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing, what, how, dan when*. Dimensi ini terdiri atas; *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration dan Change Manegement, Project Management, Environment*.



Gambar 1. Gambar Fase RUP

Gambar 1 menjelaskan 4 tahapan yang ada dalam metode RUP. Tahap- tahap dari metode RUP tersebut dapat diuraikan sebagai berikut:

1. Tahap *Inception*
Tahap *inception* adalah tahapan yang fokus pada penentuan manfaat aplikasi perangkat lunak yang harus dihasilkan. Dalam tahapan *inception* dari metode RUP langkah-langkah yang dikerjakan adalah sebagai berikut.
 - a. Mendefinisikan batasan kegiatan.
 - b. Melakukan analisis kebutuhan user.
 - c. Melakukan proses perancangan awal perangkat lunak (perancangan arsitektural).

2. Tahap *Elaboration*
Tahap *elaboration* adalah tahapan yang digunakan untuk menentukan *use case* (*set of activities*) dari perangkat lunak berikut rancangan arsitekturnya. Dalam tahapan *elaboration* dari metode RUP langkah-langkah yang dikerjakan adalah sebagai berikut.
 - a. Menspesifikasikan fitur perangkat lunak.
 - b. Menentukan *use case*.
 - c. Membuat prototipe perangkat lunak.
3. Tahap *Construction*
Tahap *construction* berisi tahapan pengimplementasian rancangan aplikasi perangkat lunak yang telah dibuat lengkap dan siap diserahkan kepada pemakai. Dalam tahapan *construction* dari metode RUP langkah-langkah yang dikerjakan adalah sebagai berikut.
 - a. Perkembangan sistem.
 - b. Melakukan sederetan iterasi. Proses yang terjadi pada setiap iterasi adalah analisa desain, implementasi dan testing.
 - c. Merilis aplikasi perangkat lunak berserta dengan dokumentasinya.
4. Tahap *Transition*
Tahap *transition* adalah tahapan untuk menyediakan sistem untuk digunakan oleh pemakai. Dalam tahapan *transition* dari metode RUP langkah-langkah yang dikerjakan adalah sebagai berikut.
 - a. Menyerahkan aplikasi perangkat lunak kepada pemakai.
 - b. Melakukan instalasi, pengujian di tempat pemakai.
 - c. Memperbaiki masalah-masalah yang muncul pada saat dan setelah pengujian

Definisi Unified Modeling Language

Penertian Unified modeling language (UML) menurut M. Salahuddin, Rossa A. S (2015 : 133) UML adalah salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Dan *UML* merupakan ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ (M. Salahuddin, Rossa A. S, 2015). Menurut Nugroho Adi dalam bukunya yang berjudul *Rekayasa Perangkat Lunak*

Berorientasi Objek Dengan Metode USDP (2010 : 7) menerangkan tentang UML yang menjadi bagian dalam bahasa pemodelan “bahwa Pemodelan (*Modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah

dipelajari dan dipahami”. *UML* lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat di akhir tahun 1980 dan pada awal tahun 1990.

Komponen Unified Modelling Language (UML)

UML menyediakan beberapa diagram visual yang berfungsi untuk menunjukkan berbagai aspek dalam sistem. Diagram mengemukakan banyak hal, penggunaan notasi yang terdefinisi baik dan ekspresif adalah penting pada proses pengembangan perangkat lunak. Ada beberapa diagram yang disediakan dalam UML menurut (M. Salahuddin, Rossa A. S, 2015) berdasarkan bukunya yang berjudul *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* yang beberapa diagramnya antara lain :

1. Diagram *use case* (*use case diagram*)

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) system yang akan dibuat, *use case* digunakan untuk mengetahui fungsi apa saja yang ada dilamam sebuah system informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. “Syarat penamaan pada *use case* adalah nama didefinisikan sesimple mungkin dan dapat dipahami” (M. Salahuddin, Rossa A. S, 2015). Komponen-komponen diagram *use case* adalah sebagai berikut :

- a. Aktor (*actor*), menggambarkan pihak – pihak yang berperan dalam sistem. jadi aktor dapat berupa orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar sisteminformasi yang akan dibuat itu sendiri
- b. *Use case*, fungsionalitas yang disediakan system untuk bertukar pesan antara unit atau actor.

2. Diagram aktivitas (*activity diagram*)

Diagram aktivitas atau *activity diagram* menggambarkan workflow atau aktifitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. “Diagram aktivitas menggambarkan aktivitas system dan bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh system” (M. Salahuddin, Rossa A. S, 2015).

3. Diagram sekuensial (*sequence diagram*)

Diagram ini menunjukkan interaksi yang terjadi antara objek. Diagram sekuensial ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi (M. Salahuddin, Rossa A. S, 2015). Adapun simbol – simbol *sequence diagram*, yaitu sebagai berikut:

4. Diagram kelas (*class diagram*)

Dalam buku Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek (M. Salahuddin, Rossa A. S, 2015) mengemukakan bahwa “Diagram kelas atau class diagram menggambarkan struktur system dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system”. Diagram kelas menunjukkan sekumpulan kelas, *interface* dan kolaborasi, dan keterhubungan. Diagram kelas ditujukan untuk pandangan statik terhadap sistem.

Java

Java adalah salah satu bahasa pemrograman yang yang digunakan untuk membuat aplikasi (Kadir A, 2013). Java dirilis pertama kali pada tahun 1995 oleh Sun Microsystems. Dan diciptakan oleh James Gosling. Java merupakan bahasa pemrograman yang berorientasikan kepada objek.



Gambar 2. Logo Bahasa Pemrograman java

“Pemrograman berorontasi objek (OOP) adalah suatu pendekatan yang memungkinkan suatu kode yang digunakan untuk menyusun program menjadi lebih mudah untuk digunakan kembali, lebih handal, dan lebih mudah dipahami” (Kadir A, 2012).

Android

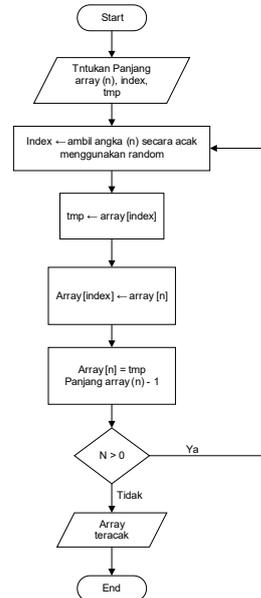
Android merupakan system operasi untuk perangkat bergerak yang dewasa ini sangat terkenal. “Awalnya android dikembangkan oleh perusahaan kecil di silicon valley yang bernama android inc. selanjutnya, Google mengambil alih system operasi tersebut pada 2005 dan mencanangkannya sebagai system operasi yang bersifat “open source” dan sebagai konsekuansinya siapapun boleh memanfaatkannya dengan gratis, termasuk dalam hal kode sumber yang digunakan untuk mnyusun system operasi tersebut” (Kadir A, 2013).

3. PERANCANGAN

Perancangan Aplikasi

Pada tahap perancangan sistem informasi bimbingan mahasiswa dengan dosen pembimbing akademik web berbasis web ini, dibangun menggunakan UML (*Unified Modelling Language*). UML adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan rancangan proses analisis dan desain berorientasi objek

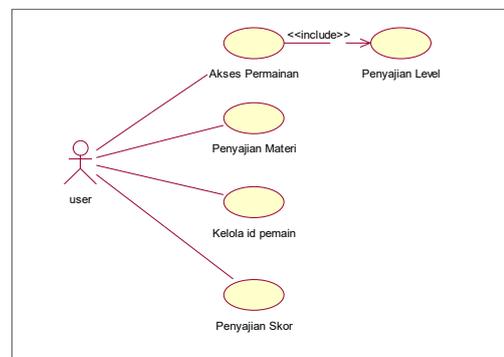
Flowchart Fisher Yates



Gambar 3. Flowchart *Algoritma Fisher Yates*

Perancangan Use Case Diagram

Use Case Diagram mendeskripsikan sistem/aplikasi, lingkungan dan relasi antara sistem/aplikasi dengan lingkungannya. Dalam aplikasi yang dibuat, user memiliki beberapa perlakuan umum yang dapat dilakukan.



Gambar 4. *Use case diagram system*

Pada gambar 4 *use case diagram* akan di deskripsikan sebagai berikut :

a. Penyajian Edukasi

- Usecase* : Penyajian materi
- Aktor : *User*
- Deskripsi : Pengguna mengakses Halaman Materi yang kemudian sitem akan menampilkan materi berupa suara berdasarkan gambar yang terpilih oleh *user*.
- Kondisi Sebelum : Halaman utama
- Kondisi Setelah : Halaman *level*

Tabel 2. Deskripsi *use case* penyajian materi (*user*)

Aktor	Sistem		
1. Pengguna mengakses halaman edukasi	2. menampilkan <i>grid</i> tampilan gambar	3. Bermain menyusun gambar <i>puzzle</i>	4. Menghitung jumlah pergerakan <i>puzzle</i>
3. Memilih <i>gambar</i>	4. Menampilkan gambar yang sesuai dengan pilihan <i>user</i>		5. Mengkalkulasikan pergerakan <i>puzzle</i> dengan nilai rata-rata pergerakan <i>puzzle</i> di level yang sama
	5. Menjalankan edukasi tentang gambar berbentuk narasi suara	6. Menggunakan <i>button solve</i>	7. Menyusun potongan <i>puzzle</i>
			8. <i>Validasi</i> susunan <i>puzzle</i>
			9. Hitung skor
			10. Menyimpan skor dalam <i>database</i>

b. Penyajian Level

Usecase : Penyajian *level*

Aktor : *User*

Deskripsi : Pengguna memilih *level* permainan untuk *grid puzzle* yang akan di tampilkan di halaman permainan.

Kondisi sebelum : Halaman pilih gambar

Kondisi setelah : Halaman permainan

Tabel 3. Deskripsi *use case* penyajian *level* (*user*)

Aktor	Sistem
1. Mengakses halaman permainan	2. Menampilkan <i>level</i> permainan dengan kategori <i>beginner</i> , <i>intermediate</i> , dan <i>advanced</i>
3. Memilih <i>level</i> permainan	4. Menampilkan halaman <i>puzzle</i> dengan <i>grid puzzle</i> sesuai <i>level</i> yang dipilih

c. Penyajian Permainan

Usecase : Akses permainan

Aktor : *User*

Deskripsi : Sistem menampilkan *grid* potongan *puzzle* berdasarkan *level* yang telah di pilih oleh pengguna, dan pengguna dapat bermain dengan menggunakan waktu dan tombol bantuan *solve* sebagai dasar untuk perolehan bintang.

Kondisi sebelum : Halaman *level*

Kondisi setelah : Halaman skor

Tabel 4. Deskripsi *use case* akses permainan (*user*)

Aktor	Sistem
1. memilih <i>level</i> permainan	2. Mengacak potongan <i>puzzle</i> menggunakan algoritma <i>fisher yates shuffle</i>

d. Kelola Id Pemain

Usecase : Kelola id pemain

Aktor : *User*

Deskripsi : User memasukan nama pengguna di awal aplikasi pertamakali di jalankan dan sistem menyimpan sebagai id pengguna aplikasi

Kondisi sebelum : Halaman *splashscreen*

Kondisi setelah : Halaman main utama

Tabel 5. Deskripsi kelola id pemain

Aktor	Sistem
1. Menjalankan aplikasi	2. Mengecek id ke dalam <i>database</i> SQLite
3. Memasukan nama <i>user</i>	4. Menambahkan id pemain ke dalam <i>database</i>
	5. Menampilkan halaman main utama

e. Penyajian Skor

Usecase : Penyajian skor

Aktor : *User*

Deskripsi : Pengguna mengakses halaman skor untuk mengetahui perolehan bintang untuk setiap gambar

Halaman sebelum: Halaman utama

Halaman setelah : Halaman detail skor

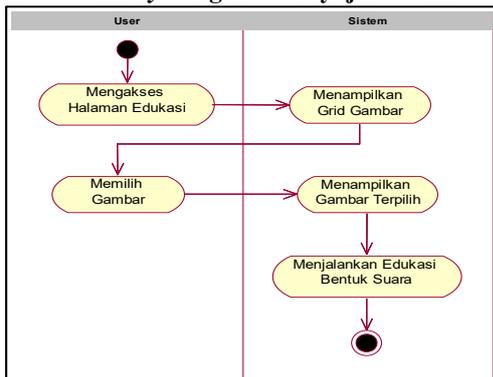
Tabel 6. Deskripsi *use case* penyajian skor (*user*)

Aktor	Sistem
1. Mengakses halaman skor, memilih gambar untuk dilihat skornya	2. Mencari data gambar ke dalam <i>database</i> , dan menampilkan data skor dalam bentuk bintang

Activity Diagram

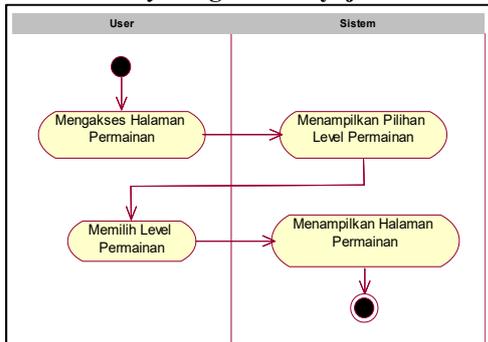
Diagram *activity* akan menampilkan aliran kerja atau aktivitas dari sebuah sistem yang ada pada aplikasi *game* edukasi berbentuk *puzzle*. Di mana aktivitas ini akan menggambarkan aktivitas sistem dan bukan yang di lakukan oleh aktor terhadap aplikasi, jadi *activity* diagram ini akan menampilkan aktivitas-aktivitas yang dilakukan oleh *system*.

a. Activity Diagram Penyajian Materi



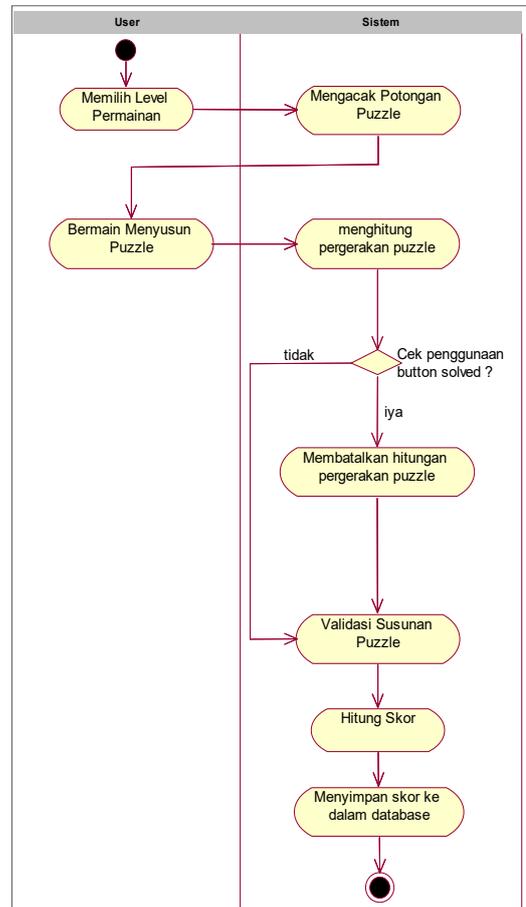
Gambar 5. Activity diagram penyajian materi edukasi

b. Activity Diagram Penyajian Level



Gambar 6. Activity diagram penyajian level

c. Activity Diagram Penyajian Permainan



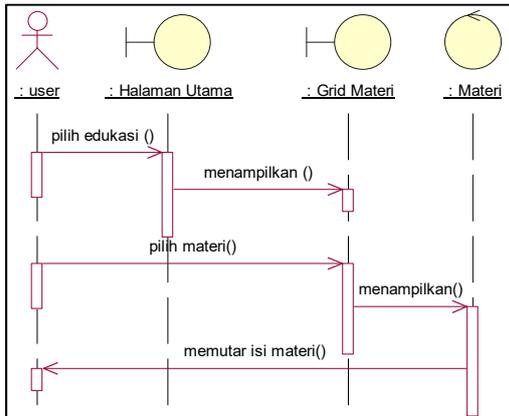
Gambar 7. Activity diagram penyajian permainan

Sequence Diagram

Sequence diagram akan menggambarkan kelakuan objek pada *use case* dengan mendefinisikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Banyaknya diagram sekuen yang harus digambarkan *minimalnya* sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah di definisikan interaksi jalannya pesan sudah mencakup pada diagram sekuen sehingga semakin banyak *use case* yang di definisikan maka diagram sekuen yang harus di buat juga semakin banyak.

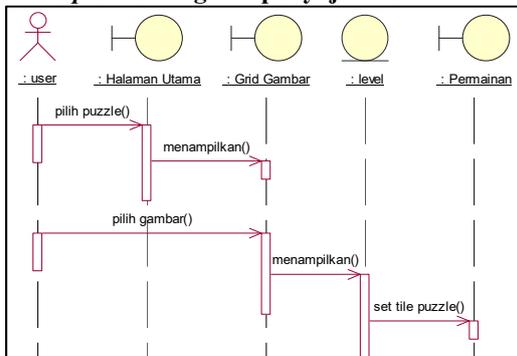
a. Sequence Diagram Penyajian Materi

Penggambaran kelakuan objek pada *use case* penyajian materi edukasi menggambarkan tentang kelakuan system ketika pengguna mengakses halaman *edukasi* tentang fungsi – fungsi yang dapat dilakukan oleh sistem



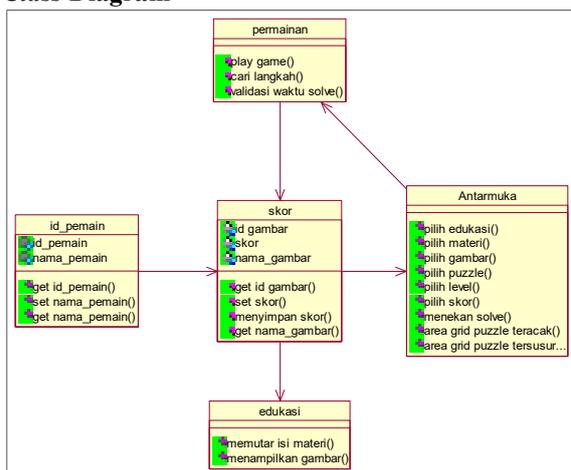
Gambar 8. *sequence* diagram penyajian materi edukasi

b. Sequence Diagram penyajian level



Gambar 9. *Sequence* diagram penyajian level

Class Diagram



Gambar 10. *Class* diagram aplikasi *game* edukasi berbentuk *puzzle*

4. HASIL DAN PEMBAHASAN

Setelah tahapan *Inception* dan *Elaboration* dilakukan, maka selanjutnya akan dilakukan tahapan *Construction* dan *Transition*. Pada tahapan *Construction* akan lebih fokus pada hasil

perancangan tampilan dan menu-menu yang diusulkan. Sementara itu, pada tahapan *Transition* lebih fokus pada proses pengujian dari aplikasi yang dirancang.

Construction

Setelah mengalami sederetan proses iterasi, pada tahapan *Construction* ini sudah dapat menghasilkan suatu aplikasi. Hasil dari aplikasi yang sudah dirancang dapat dilihat pada tampilan Gambar berikut :

1. Halaman Utama Aplikasi



Gambar 11. Halaman Utama Aplikasi

2. Halaman Utama Permainan Puzzle



Gambar 12. Tampilan Utama Menu Games Puzzle.

3. Halaman Menu Edukasi



Gambar 13. Halaman Menu Edukasi

Pengujian Perangkat Lunak

Pengujian *system* perangkat lunak adalah bagian terpenting guna mencari kesalahan-kesalahan yang ada dalam perangkat lunak yang di uji, pengujian ini dimaksudkan untuk mengetahui perangkat lunak yang di buat sudah memenuhi kriteria yang sesuai dengan perancangan awal perangkat lunak tersebut. Pengujian perangkat lunak menggunakan pengujian *black box* dan pengujian *white box*

5. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pembahasan yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut.

1. Aplikasi yang dibangun membantu anak dalam mengembangkan nilai edukasi tentang pengetahuan hewan.
2. Penerapan algoritma Fisher Yates dalam aplikasi game puzzle, sangat cocok dalam pengacakan puzzle, sehingga game lebih menantang untuk dimainkan.
3. Aplikasi ini mudah diakses dan dimainkan oleh anak SD usia 6 Tahun.

Saran

Sesuai dengan permasalahan yang ada dan setelah perancangan aplikasi ini selesai, maka diberikan beberapa saran yang dapat digunakan dalam pengembangan aplikasi di masa mendatang. Adapun saran yang ingin disampaikan yaitu sebagai berikut.

1. Dibutuhkan algoritma untuk penilaian dalam menyelesaikan potongan puzzle.
2. Perlu dikembangkan untuk system operasi selain android.

Daftar Pustaka

- [1]. Coad, Peter and Yourdon, Edward. *Object-Oriented Analysis*, Yourdon Press, 1991.
- [2]. Coad, Peter and Yourdon, Edward. *Object-Oriented Design Second Edition*, Yourdon Press, 1991.
- [3]. Fatansyah, Ir. (2007). "Basis Data". Bandung: Informatika Bandung.
- [4]. Fowler, Martin. *UML Distiled: Panduan Singkat Bahasa Pemodelan Object Standar, Edisi 3*, Penerbit Andi: Yogyakarta. 2004.
- [5]. Kruchten, Philippe, *The Rational Unified Process An Introduction, Second Edition*, John Wiley and Son Ltd. 2006.
- [6]. Nugroho Adi. *Rekayasa Perangkat Lunak Berorientasi Objek dengan metode USDP (Unified Software Development Process)*, Penerbit Andi: Yogyakarta. 2010.
- [7]. Pressman, Roger S. (2007). *Rekayasa Perangkat Lunak: pendekatan praktisi (Buku1)*. Beizer, B. (1995). *Black-Box Testing*, Wiley. Yogyakarta: Andi.
- [8]. A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures AO Ade-Ibijola - International Journal of Computer Applications, 2012 Cited by 10 - Related articles - All 7 versions