



## Game Edukasi Math & Trash Berbasis Android dengan Menggunakan Scirra Construct 2 dan Adobe Phonegap

Hardi Prasetyo<sup>1</sup>, Ida Widaningrum<sup>\*2</sup>, Indah Puji Astuti<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo

<sup>1</sup>hardi.prasetyo1@gmail.com, <sup>2\*</sup>iwidaningrum.as@gmail.com, <sup>3</sup>indahsan.0912@gmail.com

### Abstract

*Educational games can be used as an effective learning method, especially for children, because children can gain knowledge in a fun way. Here the game is designed for young children to practice arithmetic and knowledge to separate organic and inorganic wastes. This game is based on Android using Construct 2 software, design using UML and Game Development Life Cycle (GDLC). At the end of the manufacturing process, software tests or tests consist of unit tests, integration tests and system tests. From the test results, the game "Math & Trash" can be run on several versions of Android with a variety of different screen sizes. The educational game "Math & Trash" is expected to have a positive impact on children on the importance of math subjects and the importance of protecting the surrounding environment.*

*Keywords: Educational games, Construct 2, Math learning, Knowledge of organic and inorganic waste*

### Abstrak

Game edukasi dapat digunakan sebagai metode pembelajaran yang efektif, terutama untuk anak-anak, karena anak-anak bisa mendapatkan pengetahuan dengan cara yang menyenangkan. Disini dirancang game untuk anak usia dini dalam melatih berhitung dan pengetahuan memilahkan sampah organik dan anorganik. Game ini berbasis android dengan menggunakan software Construct 2, perancangan menggunakan UML dan *Game Development Life Cycle (GDLC)*. Diakhir pembuatan, dilakukan pengujian atau testing perangkat lunak yang terdiri dari unit testing, integration testing, dan system testing. Dari hasil pengujian, game "Math & Trash" mampu berjalan di beberapa versi android dengan berbagai ukuran layar yang berbeda. Game edukasi "Math & Trash", dapat memberikan dampak positif terhadap anak-anak akan pentingnya mata pelajaran matematika dan pentingnya menjaga lingkungan sekitar.

Kata kunci: Game Edukasi, Construct 2, Belajar matematika, Pengetahuan sampah organik dan anorganik.

© 2020 Jurnal RESTI

### 1. Pendahuluan

Matematika adalah mata pelajaran yang masih terkesan sangat sulit untuk dipahami, sebab itu matematika akan lebih mudah diajarkan pada anak sejak usia dini [1]. Anak-anak menganggap bahwa matematika adalah kumpulan angka-angka yang sulit untuk dipahami, padahal matematika yang diberikan masih tahapan yang sangat sederhana. Faktor lain yang menyebabkan matematika menjadi tidak menarik dan kurang dipahami anak adalah, kurangnya pemahaman dalam merancang program pengembangan matematika anak usia dini.

Semboyan kegiatan pengembangan pada anak usia dini adalah, "bermain sambil belajar dan belajar seraya

bermain", hal ini seharusnya menjadi acuan dalam pengembangan pembelajaran untuk anak. Anak akan lebih dapat menyerap proses belajar yang berhubungan dengan visualisasi dan permasalahan nyata [2]. Berdasarkan penelitian [3], salah satu media pembelajaran yang sangat mudah diterima, adalah aplikasi permainan edukasi atau game edukasi.

Game merupakan kegiatan terstruktur atau semi terstruktur, biasanya bertujuan untuk hiburan dan juga dapat digunakan sebagai sarana pendidikan [4]. Anak-anak akan cenderung mudah memahami suatu pelajaran apabila disampaikan dengan menarik dan menyenangkan. Game dapat digunakan untuk penghilang kejenuhan dalam pemberian materi pendidikan, dapat menciptakan lingkungan belajar yang

menyenangkan dan memberikan motivasi kepada anak-anak untuk belajar [5].

*Construct 2* merupakan sebuah game build berbasis HTML 5 untuk membangun suatu permainan. *Construct 2* pada dasarnya memiliki kemudahan, memiliki fitur-fitur yang mudah untuk digunakan dan dimengerti. *Construct 2* memiliki konsep Tarik dan letakkan dalam membangun suatu aplikasi game sehingga pengguna tidak perlu menuliskan baris script untuk menjalankan suatu konsep [6][7].

Game edukasi, adalah permainan yang dirancang atau dibuat untuk merangsang daya pikir termasuk meningkatkan konsentrasi dan memecahkan masalah. *Edugame* juga merupakan permainan yang digunakan dalam proses pembelajaran, permainan tersebut mengandung unsur mendidik atau nilai-nilai pendidikan[8][9]. Oleh sebab itu, pemberian materi belajar dengan metode permainan akan sangat mudah dipahami oleh anak-anak. Salah satu yang berkembang saat ini adalah aplikasi mobile phone berbasis android. Android adalah sistem operasi yang berbasis Linux untuk telepon seluler, seperti smartphone dan komputer tablet[10]. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Menurut[11], pengguna android di Indonesia meningkat 22%, hal ini dikarenakan android merupakan platform terbuka, juga memiliki kelebihan diantaranya kemudahan, interaktifitas dan user experience yang disajikan android melalui aplikasinya. Berdasarkan data dari [12], game *education* menempati urutan ketiga setelah arcade, word dan puzzle. Hal ini jelas menunjukkan bahwa game edukasi mulai berkembang dibandingkan game dengan genre lain.

Selain mengandung unsur pendidikan, game edukasi ini juga akan mengangkat pentingnya kesadaran dini dalam membuang sampah [13]. Menurut Statistik Lingkungan Hidup Indonesia 2017 kota di Indonesia menghasilkan sampah per hari paling sedikit sebanyak 100 m<sup>3</sup> [14]. Fakta ini menunjukkan masih buruknya kesadaran masyarakat Indonesia dalam membuang sampah. Oleh karena, itu perlu kesadaran akan pentingnya menjaga lingkungan hidup sejak dari dini dengan membuang sampah pada tempatnya [15].

Sudah banyak penelitian mengenai game edukasi ini, diantaranya adalah Game Math Adventure Sebagai Media Pembelajaran Matematika [1]. Game sebagai media alternatif pembelajaran untuk mengenal simbol, berhitung, mencocokkan gambar dan menyusun acak kata [2]. Game edukasi platform belajar matematika [16], dan game edukasi untuk mengenal jenis-jenis sampah [17]. Dalam penelitian ini, dirancang Game Edukasi *Math&Trash* Berbasis Android Dengan Menggunakan Scirra Construct 2 dan Phonegap. Disini, dibuat game ketrampilan berhitung dan melatih pengetahuan pengenalan lingkungan melalui

kemampuan memilahkan sampah organik dan anorganik. Bedanya dengan penelitian yang sudah ada adalah, disini dikombinasikan antara ketrampilan berhitung dengan pengetahuan tentang jenis sampah dan sasaran pengguna game ini adalah anak usia dini [18] yang masih belajar berhitung.

## 2. Metode Penelitian

Jenis penelitian yang digunakan dalam membuat aplikasi "*Math Trash*" ini adalah penelitian riset dan pengembangan (*research and development*). Menurut [19], metode penelitian dan pengembangan adalah metode yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut.

Pada penelitian ini, membangun perangkat lunak dengan metode GDLC (*Game Development Life Cycle*) [20][21]. Metode GDLC terdiri dari *preproduction*, *production*, *testing* dan *postproduction* [22][23].

Tahap *Preproduction*. Tahap pra-produksi dilakukan untuk menentukan desain game dan perencanaan seperti konsep game yang meliputi: judul game, genre game, target.

Tahap *Production*. Tahap produksi adalah terkait dengan pembuatan aspek teknis, artistik dan source codes.

Tahap Pengujian atau *Testing*. Terdiri dari validate plan dan code release. Pengujian dilakukan untuk mengetahui seberapa sempurna system yang telah dibuat[24]. Tahapan pengujian meliputi *Unit Testing*, *Integration Testing*, dan *System Testing*. Ketika system yang diuji terdapat *error* maka perlu kembali ke tahapan desain untuk memperbaiki kesalahan yang terjadi.

Tahap *Postproduction*. Terdiri dari *postmortem* dan *archive plan*. Setelah melewati tahapan pengujian, selanjutnya dipublikasikan di *PlayStore* agar dapat diunduh oleh pengguna *Smartphone* dengan platform *android*. Adapun pemeliharaan dilakukan jika sewaktu-waktu sistem yang dibuat terdapat *bug/error* ataupun upgrade sistem operasi dapat segera diatasi dan ditangani.

### 2.1. Tahap Preproduction

Game Edukasi *Math&Trash* dikembangkan untuk anak usia dini yaitu anak berusia kurang dari 6 tahun [18]. Permainan ini berkonsep tentang 2 bidang yaitu pendidikan dan lingkungan. Didalam permainan ini pengguna dituntut untuk cekatan dalam memilih sampah yang jatuh dari langit dengan nilai tertentu untuk menjawab pertanyaan yang telah diberikan. Pengguna diharuskan memasukkan sampah-sampah dengan benar antara organik dan non organik ke dalam tempat sampah dengan metode *drag and drop*. Setiap menjawab benar akan menambah skor dan lanjut ke level selanjutnya. Ketika pemain gagal untuk menjawab

dan kehabisan nyawa/waktu permainan, permainan secara otomatis akan berhenti dan menampilkan skor akhir permainan dan sistem melakukan update data.

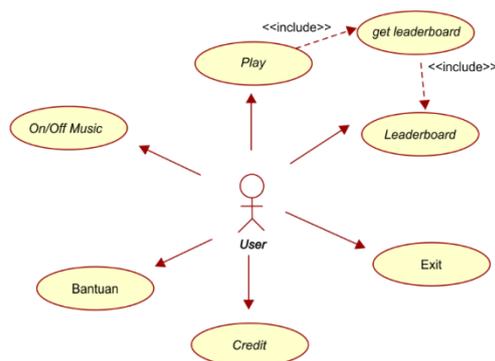
Didalam permainan ini terdapat total 30 level dengan tingkat kesulitan yang berbeda. Dan pengguna diharuskan menyelesaikan seluruh permainan untuk mendapatkan skor tertinggi.

Pemodelan yang digunakan dalam perancangan aplikasi ini adalah UML (*Unified Modelling Language*) [25][26]. Dimana diagram yang digunakan sebagai perancangan berupa *use case diagram* dan *activity diagram*. *Activity diagram* dibuat setelah perancangan *use case* dengan tujuan memberikan aliran interaksi yang terdapat dalam *use case*.

**Use case diagram**

*Use case diagram* menggambarkan seorang *user* berinteraksi dengan sistem dengan cara menentukan langkah-langkah yang akan dilakukan. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta di realisasikan oleh sebuah *collaboration*. Aktor dalam sistem perangkat lunak yang akan dikembangkan dinamakan pengguna sistem, dan aktivitas yang dilakukan sistem digambarkan dalam bentuk sejumlah *use case*.

Didalam aplikasi *Math & Trash* ini ada satu aktor sebagai pengguna aplikasi. Komunikasi aktor digambarkan dengan beberapa *use case* (gambar 1), yaitu *Play*, bantuan, *leaderboard*, *on/off music*, *credits*, dan *exit*. *Leaderboard* memiliki relasi *include* terhadap *score data*, karena ketika menjalankan fungsi *leaderboard*, *get scoreboard* akan dijalankan terlebih dahulu untuk mendapatkan skor permainan yang tersimpan dalam web storage. *Use case get scoreboard* memiliki relasi *include* dengan *use case* mulai bermain, karena ketika bermain skor akan tersimpan didalam *scoreboard*.



Gambar 1. Use Case Diagram Math&Trash

Penjabaran *use case diagram* diuraikan lebih lanjut pada Tabel 1.

**Activity Diagram**

*Activity Diagram* (gambar 2) menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang. Masing-masing alir berawal, percabangan yang terjadi, dan bagaimana akhir aktivitas. Diagram aktivitas yang disediakan UML melengkapi *use case* yang telah dibuat sebelumnya dengan memberikan representasi grafis dari alir-alir interaksi di dalam suatu skenario yang sifatnya spesifik [27].

Tabel 1. Tabel scenario

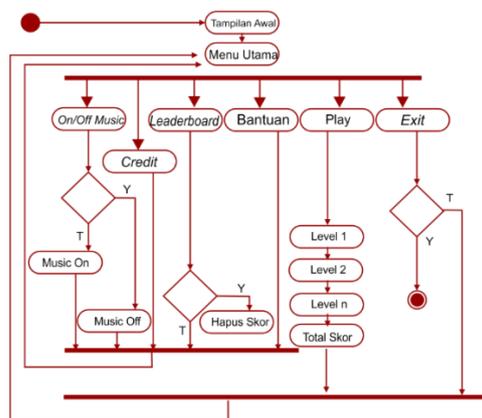
Aksi Aktor	Reaksi Sistem
<b>Skenario Menjalankan Aplikasi</b>	
Pengguna Menjalankan Aplikasi	Menuju ke menu utama aplikasi
<b>Skenario Membuka Play</b>	
Pengguna Menekan Play	Menuju ke halaman permainan
<b>Skenario Memainkan Game</b>	
Memainkan Game	<ul style="list-style-type: none"> <li>Menampilkan permainan dan langsung masuk ke level 1</li> <li>Jika menang menampilkan skor</li> <li>Menampilkan pesan menang dan langsung menuju level 2</li> </ul>
Menekan Home	<ul style="list-style-type: none"> <li>Update skor pada <i>Leaderboard</i></li> <li>Kembali ke menu utama</li> <li>Jika tidak, kembali melanjutkan permainan</li> </ul>
Menekan Exit	<ul style="list-style-type: none"> <li>Update skor pada <i>Leaderboard</i></li> <li>Keluar dari permainan</li> <li>Jika tidak, kembali melanjutkan permainan</li> </ul>
<b>Skenario Membuka Leaderboard</b>	
Menekan <i>Leaderboard</i>	Menampilkan daftar 10 besar pemain dengan skor tertinggi
<b>Skenario Mereset Data Score</b>	
Menekan <i>Reset Score</i>	<ul style="list-style-type: none"> <li>Menampilkan pesan peringatan</li> <li>Jika ya. Menghapus semua daftar pemain dengan skor tertinggi</li> <li>Menampilkan pesan peringatan</li> <li>Jika tidak, tidak menghapus daftar pemain dengan skor tertinggi</li> </ul>
<b>Skenario On/Off Music/Sound</b>	
Menekan <i>on/off</i> musik / <i>sound</i>	Music dan <i>sound</i> berhenti
	Music dan <i>sound</i> berbunyi kembali
<b>Skenario Membuka Bantuan</b>	
Menekan/ bantuan	Menampilkan tata cara bermain dan menjelaskan jenis-jenis sampah
<b>Skenario Membuka Credits</b>	
Menekan <i>Credits</i>	Menampilkan nama pembuat aplikasi
<b>Skenario Keluar Aplikasi</b>	
Keluar Aplikasi	Proses menutup aplikasi

Aliran aktivitas berawal dari tampilan awal kemudian masuk ke menu utama. Di dalam menu utama terdapat beberapa menu lainnya seperti *play*, *on/off music*, *leaderboard*, bantuan, dan *exit*. Ketika memilih menu *on/off music* secara otomatis sistem akan melakukan *decision* mematikan musik atau menghidupkan musik. Aktivitas sistem pada *leaderboard* akan menampilkan skor yang telah tersimpan dan memiliki *decision* untuk mereset data skor. Aktivitas pada menu bantuan akan menampilkan cara bermain dan perbedaan asset

sampah yang akan digunakan didalam permainan dan kembali ke menu utama. Aktivitas di menu *play* terdapat 30 level dengan tingkat kesulitan berbeda. Setelah pengguna masuk di permainan sistem *game* akan menampilkan permainan yang sesungguhnya, yaitu menampilkan aset-aset yang digunakan dalam *game*. Didalam permainan ini akan menampilkan sampah organik dan nonorganik yang jatuh dari langit secara acak dan terus menerus. Didalam sampah tersebut terdapat nilai yang acak yang digunakan sebagai menjawab pertanyaan yang keluar di permainan. Pengguna akan berinteraksi dengan melakukan *drag and drop* untuk diletakkan pada tempat sampah. Pengguna diharuskan memilah antara sampah organik dan nonorganik dengan benar, jika pemain salah dalam memasukkan sampah maka bar waktu akan berkurang 5 detik. Jika pengguna benar dalam memasukkan sampah maka skor akan bertambah 1.

Setiap interaksi yang dilakukan pengguna dengan aset sampah ini akan dikumpulkan nilai tertentu yang digunakan untuk menjawab soal. Setiap pengguna mencapai nilai benar, secara otomatis akan menambah skor. Di setiap level terdapat beberapa pertanyaan berupa angka yang harus dicapai pengguna untuk dapat lanjut ke step yang selanjutnya. Ketika pengguna gagal atau kalah dalam permainan maka akan ditampilkan halaman *game over*. Permainan akan diakhiri dan sistem akan melakukan update data berupa skor. Setiap level memiliki asset yang sama tetapi dengan tingkat kesulitan yang berbeda

Aliran aktivitas di menu *exit* yaitu sistem akan memberikan peringatan berupa akan keluar dan batal keluar. Ketika pengguna menekan keluar sistem akan keluar dari permainan.



Gambar 2. Activity Diagram Game edukasi “Math & Trash”

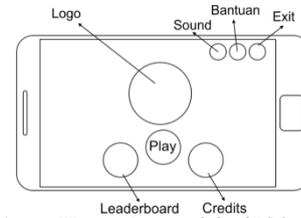
### Perancangan Antarmuka Sistem (User Interface)

Perancangan antar muka merupakan tampilan awal dari aplikasi. *User Interface* merupakan jembatan pesan antara *user* dan sistem. Pengguna akan mendapatkan output dari *user interface*

Desain antar muka harusnya bersifat menarik agar pengguna merasa nyaman dan tertarik untuk terus

menggunakan aplikasi. Adapun desain antar muka pada *game* “*Math & Trash*” antara lain sebagai berikut:

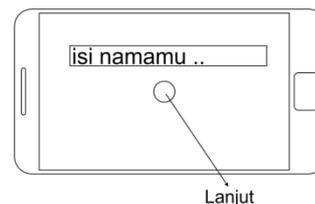
#### 1. Halaman Utama



Gambar 3. Halaman Utama Game edukasi “Math&Trash”

Pada Halaman Utama ini terdapat 7 sub-menu (gambar 3). Menu *Play* digunakan untuk memulai *game Math&Trash*. Menu *Leaderboard* digunakan untuk melihat daftar 10 skor tertinggi dalam permainan. Menu *Credit* untuk melihat membuat permainan. Menu *Bantuan* digunakan untuk membantu *user* dalam memainkan permainan. Didalam menu bantuan ini terdapat tutorial memainkan permainan *Math Trash* dan macam-macam jenis sampah organik dan non organik. Menu *On/Off Music* Digunakan untuk menghidupkan dan mematikan Music. Menu *Exit* digunakan untuk keluar dari permainan.

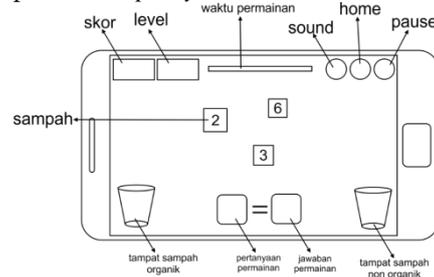
#### 2. Tampilan Mengisi Nama



Gambar 4. Halaman Mengisi Nama

Pengguna diharuskan mengisi nama untuk dapat masuk ke *Gameplay* (gambar 4). Nama tersebut akan dimasukkan di *leaderboard* untuk mengetahui pemain dengan nilai skor terbanyak.

#### 3. Tampilan Gameplay

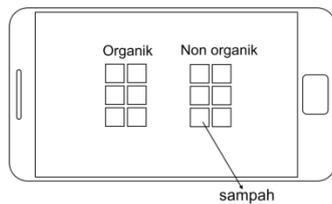


Gambar 5. Halaman gameplay “Math & Trash”

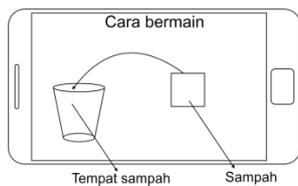
Setelah mengisi nama *user* akan masuk ke halaman *gameplay* (gambar 5). Halaman *gameplay* ini adalah halaman utama dari permainan. Halaman ini berisi tentang item-item sampah yang jatuh dari atas. Didalam sampah yang jatuh tersebut terdapat nilai-nilai acak yang digunakan untuk menjawab pertanyaan yang telah ada. Sampah yang jatuh terdiri dari sampah organik dan

sampah non organik. *User* tinggal memilih item sampah lalu memindahkan ke tempat keranjang dengan cara *drag and drop* sesuai dengan jenis sampah. Sampah organik bernilai *plus (+)* dan sampah non organik bernilai *minus (-)*. Setiap memasukkan sampah dengan jenis yang benar maka skor akan bertambah. Jika pertanyaan benar secara otomatis sistem akan menambah skor. Setiap salah dalam memasukkan sampah maka waktu akan berkurang 5 detik. Jika pertanyaan tidak dapat dijawab sampai waktu habis maka permainan berakhir. Didalam halaman *gameplay* ini terdapat menu *On/Off music* untuk menghidupkan atau mematikan musik, menu *home* untuk kembali ke menu awal, dan *exit* untuk keluar dari permainan.

#### 4. Tampilan Bantuan



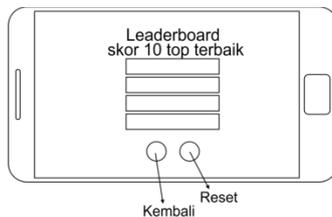
Gambar 6. Halaman Bantuan 1 “Math & Trash”



Gambar 7. Halaman Bantuan 2 “Math & Trash”

Pada halaman bantuan (gambar 6 dan 7) berisi tentang jenis-jenis sampah yang akan digunakan dalam permainan meliputi: sampah organik dan sampah non organik. Pada halaman ini pula terdapat tatacara bermain *game Math&Trash*, yaitu cara bermain dan cara menjawab benar.

#### 5. Tampilan Leaderboard



Gambar 8. Halaman Scoreboard “Math & Trash”

Halaman Scoreboard (gambar 8) berisi nilai tertinggi selama bermain. Nilai akan otomatis mengalami perubahan jika *user* berhasil mendapatkan nilai lebih tinggi. Selain itu terdapat menu *reset* untuk menghapus ulang skor dan menu *kembali* untuk kembali ke halaman utama.

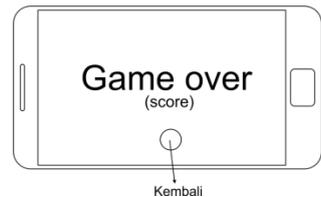
#### 6. Tampilan Credit



Gambar 9. Halaman Credits “Math & Trash”

Halaman ini berisi tentang informasi diri pembuat *game* (gambar 9).

#### 7. Tampilan Game Over



Gambar 10. Halaman Game Over “Math & Trash”

Halaman ini akan muncul ketika *user* gagal menjawab pertanyaan yang ada karena kehabisan waktu (gambar 10). Pada halaman *game over* ini menampilkan jumlah skor terakhir yang akan diperbarui di menu *Leaderboard*.

#### Teknik Analisis Data

Analisis data pada penelitian ini menggunakan skala pengukuran Likert[28]. Skala pengukuran Likert digunakan untuk menghitung variable yang diujikan, yaitu *functionality* dan *portability*. Data kuantitatif akan diubah menjadi data kualitatif dengan skala Likert. Variabel *functionality* dilakukan berdasarkan uji *whitebox* dan *blackbox*. Hasil instrumen dihitung rata-rata dengan rumus:

1. Rumus perhitungan rata-rata instrumen:

$$x = \frac{\sum x}{n}$$

Keterangan:  $x$  = Skor rata-rata  
 $\sum x$  = Skor total item  
 $n$  = Jumlah item

2. perhitungan persentase skor ditulis dengan rumus berikut:

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

Hasil dari presentase tersebut kemudian dicocokkan dengan predikat skala Likert (tabel 2).

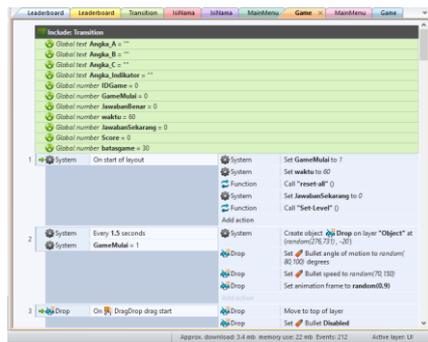
Tabel 2. Skala Penilaian Likert

No	Presentase	Interpretasi
1	0% - 20%	Sangat Tidak Layak
2	21% - 40%	Tidak Layak
3	41% - 60%	Cukup Layak
4	61% - 80%	Layak
5	81% - 100%	Sangat Layak

#### 2.2. Tahap Production

Desain yang telah dirancang kemudian diaplikasikan kedalam kode melalui event-event yang ada di aplikasi

Construct 2 [29]. Adapun implementasi logika dan event yang dilakukan seperti pada gambar 11 (a – c).



Gambar 11 a. Implementasi Event1

Pada gambar 11 a., terlihat tampilan dimana disana terdapat beberapa global variabel yang digunakan mulai dari global variabel a,b,c yang digunakan sebagai output dari pertanyaan, global variabel indikator sebagai penanda pengurangan atau penambahan, lalu ada global variabel waktu sebagai batas waktu permainan, dan global variabel batas game sebagai batas akhir dari permainan. Awal permainan pada event ini adalah dimulai dari ketika mulai masuk di event game, maka *game* mulai ke ID 1 lalu melakukan setting waktu 60 detik, lalu objek akan muncul setiap 1.3 detik secara acak mulai dari sampah organik maupun nonorganik. Lalu ketika objek di tarik atau *drag* otomatis objek akan hancur ketika dilepas atau *drop*.



Gambar 11 b. Implementasi Event2

Pada gambar 11 b., terlihat tampilan dimana disana terdapat ID game sebagai penanda dari setiap level pada permainan, dimulai dari ID game 1 sebagai level 1 dimana set *Angka\_A* 3 adalah pertanyaan nya dan *Jawaban Benar* 3 sebagai jawaban yang harus di dapatkan oleh user. Pada level 6 terdapat perbedaan dengan level 1 sampai 5, pada level ini terdapat kesulitan yang bertambah berupa penambahan indikator. Terlihat pada level 6 *Angka\_A* adalah 2 lalu terdapat indikator + dan *Angka\_C* 8, user harus mencari berapa nilai dari *Angka\_B* supaya mendapatkan jawaban benar.

Pada gambar 11 c., terlihat tampilan implementasi antar muka aplikasi menggunakan Construct 2. Implementasi antarmuka *game "Math&Trash"* dimulai dengan Gambar 12, sebagai halaman awal atau halaman Utama. Pada Halaman Utama ini terdapat 7 sub-menu. Menu *Play* digunakan untuk memulai *game*

*Math&Trash*. Menu *Leaderboard* digunakan untuk melihat daftar 10 skor tertinggi dalam permainan. Menu *Credit* untuk melihat pembuat permainan. Menu *Bantuan* digunakan untuk membantu *user* dalam memainkan permainan. Didalam menu bantuan ini terdapat tutorial memainkan permainan *Math Trash* dan macam-macam jenis sampah organik dan non organik. Menu *On/Off Music* Digunakan untuk menghidupkan dan mematikan Music. Menu *Exit* digunakan untuk keluar dari permainan.



Gambar 11 c. Implementasi Event3



Gambar 12. Halaman Utama "Math & Trash"

Halaman selanjutnya, seperti terlihat pada gambar 13, adalah tampilan Mengisi Nama. Pengguna diharuskan mengisi untuk dapat masuk ke *Gameplay*. Nama akan dimasukkan di *leaderboard* untuk mengetahui pemain dengan nilai skor terbanyak.



Gambar 13. Halaman Masukkan Nama "Math & Trash"

Setelah mengisi nama *user* akan masuk ke halaman *gameplay* seperti terlihat pada gambar 14. Halaman *gameplay* ini adalah halaman utama dari permainan. Halaman ini berisi tentang item-item sampah yang jatuh dari atas. Didalam sampah yang jatuh tersebut terdapat nilai-nilai acak yang digunakan untuk menjawab pertanyaan yang telah ada. Sampah yang jatuh terdiri dari sampah organik dan sampan non organik. *User* tinggal memilih item sampah lalu memindahkan ke tempat keranjang dengan cara *drag and drop* sesuai dengan jenis sampah. Sampah organik bernilai *plus (+)* dan sampah non organik bernilai *minus (-)*. Setiap memasukkan sampah dengan jenis

yang benar maka skor akan bertambah. Jika pertanyaan benar secara otomatis sistem akan menambah skor. Setiap salah dalam memasukkan sampah maka waktu akan berkurang 5 detik. Jika pertanyaan tidak dapat dijawab sampai waktu habis maka permainan berakhir. Didalam halaman *gameplay* ini terdapat menu *On/Off music* untuk menghidupkan atau mematikan musik, menu *home* untuk kembali ke menu awal, dan *exit* untuk keluar dari permainan.



Gambar 14. Halaman *gameplay* "Math & Trash"

Gambar 15, 16, 17 memperlihatkan tampilan bantuan. Pada halaman ini berisi tentang jenis-jenis sampah yang akan digunakan dalam permainan meliputi: sampah organik dan sampah non organik. Pada halaman ini pula terdapat tatacara bermain *game Math&Trash*, yaitu cara bermain dan cara menjawab benar.



Gambar 15. Halaman Bantuan 1 "Math & Trash"



Gambar 16. Halaman Bantuan 2 "Math & Trash"



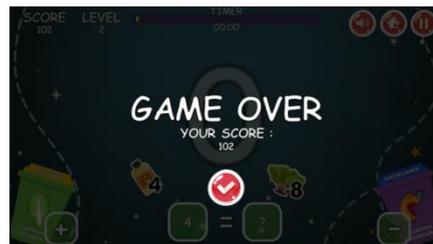
Gambar 17. Halaman Bantuan 4 "Math & Trash"

Gambar 18 merupakan halaman *Leaderboard*, menampilkan nilai tertinggi selama bermain. Nilai akan otomatis mengalami perubahan jika *user* berhasil mendapatkan nilai lebih tinggi. Selain itu terdapat

menu *reset* untuk menghapus ulang skor dan menu kembali untuk kembali ke halaman utama.



Gambar 18. Halaman *Leaderboard* "Math & Trash"



Gambar 19. Halaman *Game Over* "Math & Trash"

Gambar 19 menampilkan keterangan *Game Over*, halaman ini akan muncul ketika *user* gagal menjawab pertanyaan yang ada karena kehabisan waktu. Pada halaman ini, akan muncul jumlah skor terakhir yang akan diperbarui di menu *Leaderboard*.

### Teknik Analisis Data

Analisis data pada penelitian ini menggunakan skala pengukuran Likert. Skala pengukuran Likert digunakan untuk menghitung variable yang diujikan, yaitu *functionality* dan *portability*. Data kuantitatif akan diubah menjadi data kualitatif dengan skala Likert. Variabel *functionality* dilakukan berdasarkan uji *whitebox* dan *blackbox*. Hasil instrumen nantinya akan dihitung rata-rata dengan rumus:

1. Rumus perhitungan rata-rata instrumen:

$$x = \frac{\sum x}{n}$$

Keterangan:  $x$  = Skor rata-rata  
 $\sum x$  = Skor total item  
 $n$  = Jumlah item

2. Rumus perhitungan persentase skor ditulis dengan rumus berikut:

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

Hasil dari presentase tersebut kemudian dicocokkan dengan predikat skala Likert.

No	Presentase	Interpretasi
1	0% - 20%	Sangat Tidak Layak
2	21% - 40%	Tidak Layak
3	41% - 60%	Cukup Layak
4	61% - 80%	Layak
5	81% - 100%	Sangat Layak

**Build Aplikasi**

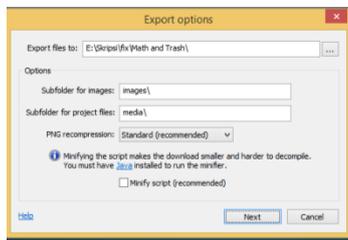
Setelah fungsi-fungsi program sudah sesuai dengan fungsinya masing-masing, *project* akan diubah menjadi *Mathandtrash.apk*. proses pertama yaitu *exportproject* adalah sebagai berikut:

Klik *File – export project* kemudian pilih *Cordova*, dan klik *next*(gambar 20).



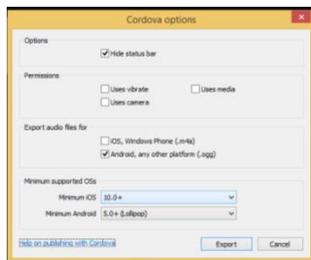
Gambar 20. *Exporting project* menggunakan *Cordova*

Pilih tempat menyimpan *file project*, kemudian klik *next* (gambar 21).



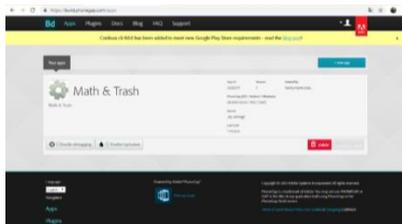
Gambar 21. Pilihan tempat menyimpan *file*

Centang pada *hide status bar* dan centang pada android karena aplikasi berbasis android (gambar 22), akhirnya file berhasil di *export*.



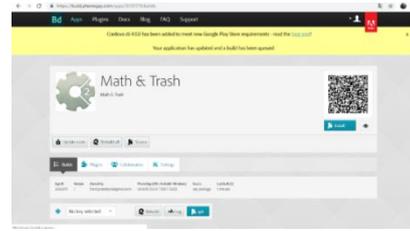
Gambar 22. Pilihan *options cordova*

Selanjutnya masuk situs adobe *Phonegap*, login kemudian *Upload file* hasil *exporting project* dengan menekan *upload a.zip file* (gambar 23).



Gambar 23. Halaman *upload project*

Klik tombol *.apk* untuk menjadikan *project* menjadi *mathandtrash.apk* (gambar 24).



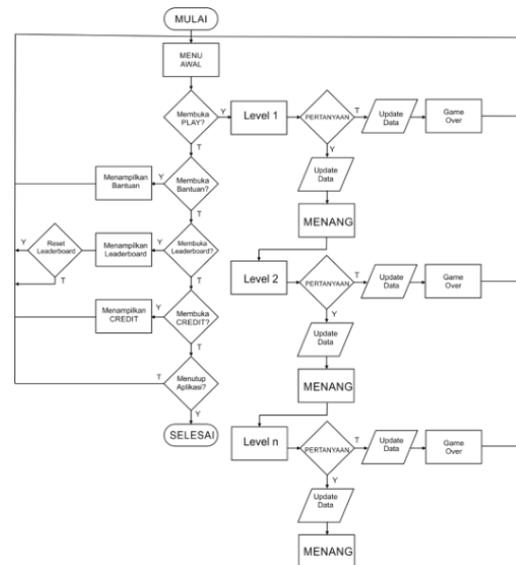
Gambar 24. Membuat *mathandtrash.apk*

**2.3. Tahap Pengujianatau Testing**

**Unit Testing**

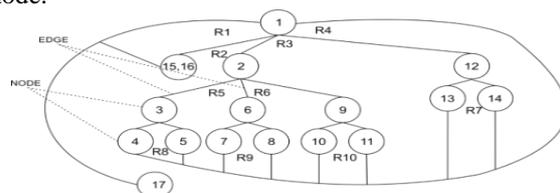
*Unit testing* merupakan proses pengujian yang berfokus pada kode program. Tujuannya untuk memeriksa bahwa semua unit-unit event sudah melakukan apa yang telah dirancang. *Unittesting* berfokus pada upaya verifikasi terhadap unit terkecil dari perancangan perangkat lunak, yaitu pada logika pemrosesan internal dan struktur data didalam komponen [30]. Pengujian menggunakan metode *white box testing*.

Pengujian *white box* merupakan pengujian perangkat lunak yang berfokus pada struktur kontrol program. Langkah-langkah pengujiannya sebagai berikut: Pertama menentukan grafik alir (*flowgraph*). *Flowgraph* merupakan notasi sederhana dari aliran control (gambar 26) yang didasarkan pada *flowchart* program (gambar 25).



Gambar 25. *FlowchartMath & Trash*

Menurut Pressman [30], urutan kotak-kotak proses dan berlian-berlian keputusan bisa dipetakan dalam satu node. Sedangkan sebuah *edge* harus berhenti di sebuah node.



Gambar 26. *Flowgraph "Math & Trash"*

Keterangan Node *Flowgraph Math & Trash*:

- (1) Mulai
- (2) Menu Play
- (3) Kondisi IF pada permainan level satu. Jika menang menuju node 4 dan jika kalah menuju node 5.
- (6) Kondisi IF pada permainan level dua. Jika menang menuju node 7, dan jika kalah menuju node 8.
- (9) Kondisi IF pada permainan level n. Jika menang menuju node 10, dan jika kalah menuju node 11.
- (12) Kondisi IF pada reset skor. Jika ya maka menuju node 13, dan jika tidak menuju node 14.
- (15) Membuka bantuan
- (16) Membuka *Credit*
- (17) Keluar

Langkah kedua dari pengujian white box testing, adalah menghitung *Cyclomatic Complexity*. *Cyclomatic Complexity* adalah metric perangkat lunak yang menyediakan ukuran kuantitatif dari suatu program. Nilai yang dihitung mendefinisikan jumlah *independent Path*, serta menyediakan batas atas untuk memastikan semua pernah diuji.

Berdasarkan teori *graph* [30], kompleksitas sistematis dapat dihitung dengan melalui jumlah region atau daerah pada *flowgraph*. Menggunakan persamaan  $V(G) = E - N + 2$ , dimana E adalah jumlah *edge* dan N adalah jumlah node. Persamaan yang lain adalah  $V(G) = P + 1$ , dimana P adalah jumlah *node*. Berdasarkan *Flowgraph*, diketahui  $Edge = 24$  dan  $Node$

= 16, maka persamaan  $V(G) = E - N + 2 = 24 - 16 + 2 = 10$ . Sehingga nilai kompleksitas sistematisnya sebesar 10. Dengan demikian harus ada 10 *test case* agar semua dapat dieksekusi.

Langkah ketiga, menentukan *Independent Path*. *Independent Path* adalah jalur yang dilalui program dalam merepresentasikan suatu tindakan. *Independent Path* ini akan digunakan untuk melakukan uji *test case*. Berdasarkan *Flowgraph* pada gambar 21, dihasilkan 10 path, dengan:

- Path 1 : 1-17
- Path 2 : 1-15-16-1-17
- Path 3 : 1-2-3-4-1-17
- Path 4 : 1-2-3-5-1-17
- Path 5 : 1-2-6-7-1-17
- Path 6 : 1-2-6-8-1-17
- Path 7 : 1-2-9-10-1-17
- Path 8 : 1-2-9-11-1-17
- Path 9 : 1-12-13-1-17
- Path 10 : 1-12-14-1-17

Nilai ini, sesuai dengan perhitungan yang dilakukan pada kompleksitas sistematis

Langkah terakhir adalah membuat *Test Case*. Menurut Standart IEEE [24], *test case* adalah kumpulan dari input tes, kondisi yang akan dieksekusi, dan hasil yang diharapkan. Ini merupakan pengujian jalur dasar yaitu membuat *test case* untuk masing-masing basis set. *Test case* dari game *Math & Trash*, diuraikan pada Tabel 2.

Tabel 2. Uji *Test case*

No	Nama	Input	Output	Ketercapaian
1	<i>Test Case 1</i>	Keluar dari aplikasi dan menampilkan konfirmasi untuk keluar	Muncul kotak konfirmasi keluar	Tercapai
2	<i>Test Case 2</i>	Menampilkan menu Bantuan dan menu <i>Credit</i>	Muncul Menu Bantuan dan Menu <i>Credit</i>	Tercapai
3	<i>Test Case 3</i>	Menampilkan level 1 dan pertanyaan pertama dengan benar dan kondisi menang	Muncul pertanyaan masuk ke permainan dengan menampilkan tampilan menang	Tercapai
4	<i>Test Case 4</i>	Menampilkan level 1 dan pertanyaan pertama dengan benar dan kondisi kalah	Muncul pertanyaan dan masuk ke permainan dengan menampilkan tampilan kalah	Tercapai
5	<i>Test Case 5</i>	Menampilkan level 2 dan pertanyaan kedua dengan benar dan kondisi menang	Muncul pertanyaan dan masuk ke permainan dengan menampilkan tampilan menang	Tercapai
6	<i>Test Case 6</i>	Menampilkan level 2 dan pertanyaan kedua dengan benar dan kondisi kalah	Muncul pertanyaan dan masuk ke permainan dengan menampilkan tampilan kalah	Tercapai
7	<i>Test Case 7</i>	Menampilkan level n dan pertanyaan ke-n dengan benar dan kondisi menang	Muncul pertanyaan dan masuk ke permainan dengan menampilkan tampilan menang	Tercapai
8	<i>Test case 8</i>	Menampilkan level n dan pertanyaan ke-n dengan benar dan kondisi kalah	Muncul pertanyaan dan masuk ke permainan dengan menampilkan tampilan kalah	Tercapai
9	<i>Test case 9</i>	Daftar skor tertinggi yang diperoleh di tampilkan pada <i>leaderboard</i> dapat direset dengan kondisi Ya	Tampilan <i>leaderboard</i> menjadi 0	Tercapai
10	<i>Test case 10</i>	Daftar skor tertinggi yang diperoleh di tampilkan pada <i>leaderboard</i> dapat direset dengan kondisi Tidak	Tampilan <i>leaderboard</i> batal mereset skor	Tercapai

Berdasarkan hasil pengujian pada tabel, didapat presentase ketercapaian sebagai berikut:

$$\begin{aligned} \text{Tercapai} &= \frac{10}{10} \times 100\% = 100\% \\ \text{Gagal} &= \frac{0}{10} \times 100\% = 0\% \end{aligned}$$

Berdasarkan nilai kuantitatif yang didapatkan dan dikonversikan menjadi kualitatif. Maka hasil pengujian *White box* yang dipaparkan adalah Sangat Baik.

**Integration Testing**

Tabel 3. Uji *Black Box* Memulai Aplikasi

Nama pengujian	Tujuan	Skenario	Hasil yang diharapkan	Status
Menjalankan aplikasi	Mengetahui aplikasi berjalan tanpa <i>error</i>	User menjalankan aplikasi	Aplikasi dapat berjalan dengan baik, menampilkan menu utama dan mengaktifkan musik <i>game</i>	Sesuai
Menekan ikon <i>sound</i>	Mengetahui reaksi musik jika ikon <i>sound</i> ditekan	User menekan ikon <i>sound</i>	Musik <i>game</i> nonaktif	Sesuai
Membuka <i>Play</i>	Agar pemain dapat memulai permainan	User kembali menekan ikon <i>sound</i>	Musik <i>game</i> aktif kembali	Sesuai
Memainkan <i>Game</i>	Pemain memainkan permainan sesuai dengan aturan	Membuka <i>Play</i> , memulai ke permainan	Masuk ke permainan	Sesuai
Data Skor	Aplikasi mampu menyimpan data skor di <i>database</i>	Memainkan <i>game</i> untuk dimenangkan	<i>Item</i> sampah muncul secara acak dan memiliki nilai yang berbeda-beda	Sesuai
Reset Skor	Pemain dapat mengatur ulang skor di <i>leaderboard</i>	Memenangkan pertandingan agar dapat membuka level selanjutnya dan mendapatkan nilai	Data skor dapat tersimpan ketika permainan berakhir atau pemain keluar dari permainan	Sesuai
Keluar Aplikasi	Agar pemain dapat keluar dari aplikasi	Menekan tombol <i>reset</i> di <i>leaderboard</i>	Skor dapat diatur ulang	Sesuai
		Menekan tombol keluar pada aplikasi	Aplikasi dapat menutup dengan sempurna	Sesuai

Berdasarkan hasil pengujian, dapat diketahui presentase kecapaian sebagai berikut:

$$\begin{aligned} \text{Tercapai} &= \frac{7}{7} \times 100\% = 100\% \\ \text{Gagal} &= \frac{0}{7} \times 100\% = 0\% \end{aligned}$$

Berdasarkan hasil pengujian *blackbox* diketahui keseluruhan sistem yang dikembangkan adalah memiliki fungsionalitas yang Sangat Baik.

**System Testing**

Tahapan ketiga pengujian perangkat adalah *system testing*[32]. Pengujian dilakukan menggunakan tes secara langsung dengan perangkat, untuk mendapatkan hasil aspek *portability*.

Pengujian menggunakan 3 perangkat yang berbeda-beda berdasarkan system operasi dan ukuran layar. Hasil pengujian aspek *portability* diperlihatkan pada Tabel 4.

Tabel 4. Hasil pengujian *Portability*

No	Smartphone	Versi Android	Ukura layar	Hasil
1	Smartphone Xiaomi Redmi 4A Prime	Android 7.1.2 (Nougat)	720 x 1280 pixels, 5.0 inches	Aplikasi berhasil diinstall dan dapat berjalan dengan baik (dokumentasi pada gambar)
2	Smartphone Xiaomi Redmi S2	Android 8.1 (Oreo)	720 x 1440 pixels, 5.99 inches	Aplikasi berhasil dinstal dan dapat berjalan dengan baik (dokumentasi pada gambar)

Tahapan kedua pengujian perangkat lunak kali ini adalah *Integration testing*. Pada pengujian tahap ini akan menggunakan metode *Black Box* untuk menguji aspek *fungsional* perangkat lunak (tabel 3). Pengujian *blackbox* merupakan pengujian terintegrasi yang dilakukan dengan cara mengamati hasil eksekusi melalui data uji dan memeriksa fungsi dari perangkat lunak[31].

3	Smartphone Xiaomi Redmi 6A	Android 8.1 (Oreo)	720 x 1440 pixels, 5.45 inches	Aplikasi berhasil diinstal dan dapat berjalan dengan baik (dokumentasi pada gambar)
---	----------------------------	--------------------	--------------------------------	-------------------------------------------------------------------------------------

Pengujian *portability* menggunakan *smartphone* android. *Smartphone* yang digunakan menggunakan versi Nougat dan Oreo. Ukuran layar yang digunakan masing-masing *smartphone* berbeda, yaitu *smartphone* layar 5 inchi, 5.45 inchi, dan 5.90 inchi.

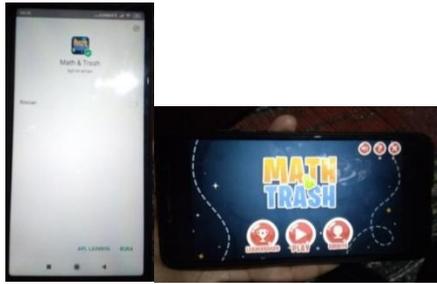
Hasil dari pengujian ini didapat bahwa aplikasi mampu diinstal dan dijalankan dengan baik

Dokumentasi pengujian aplikasi pada *smartphone* Xiaomi Redmi 4A Prime dapat dilihat pada gambar 27.

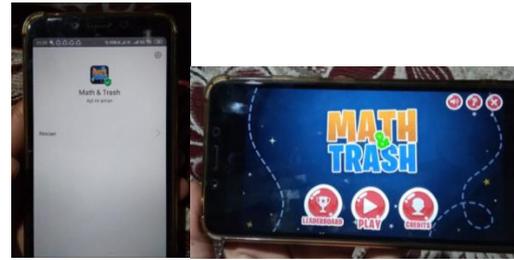


a. Instalasi Aplikasib Instalasi berhasil  
 Gambar 27. *Smartphone* Xioami Redmi 4A.

Dokumentasi pengujian aplikasi pada *Smartphone* Xiaomi Redmi S2 dapat dilihat pada gambar 28.



a. instalasi Aplikasib. Instalasi berhasil  
 Gambar 28. *Smartphone* Xioami Redmi S2.



a. Instalasi Aplikasib Instalasi berhasil  
 Gambar 29. Instalasi Aplikasi pada *Smartphone* Xiaomi Redmi 6A.

Dokumentasi pengujian aplikasi pada *Smartphone* Xiaomi Redmi 6A dapat dilihat pada gambar 29.

Hasil pengujian *portability* aplikasi secara umum ditampilkan dalam Tabel 5.

Tabel 5. Hasil Pengujian *Portability* Aplikasi

Indikator	Skenario	Hasil yang diharapkan	Ketercapaian		Keterangan
			Ya	Tidak	
Installability (Kemampuan untuk Diinstal)	Kemudahan install aplikasi	Aplikasi "Math&Trash" dapat diinstal dengan mudah	Ya		Berjalan baik tanpa ada pesan kesalahan
	Keberhasilan dalam instalasi	Aplikasi "Math&Trash" dapat diinstal dengan baik	Ya		Berjalan baik tanpa ada pesan kesalahan
Adaptability (Kemampuan untuk Beradaptasi)	Instalasi di beberapa versi android	Aplikasi "Math&Trash" dapat diinstal di beberapa versi android	Ya		Berjalan baik tanpa ada pesan kesalahan
	Instalasi di beberapa ukuran layar	Aplikasi "Math&Trash" dapat diinstal di berbagai ukuran layar berbeda	Ya		Berjalan baik tanpa ada pesan kesalahan

Kesimpulan dari hasil pengujian aspek *portability* dari Aplikasi "Math & Trash" memiliki portabilitas yang Sangat baik.

#### 2.4. Tahap Postproduction

##### Revisi Produk

Setelah melakukan pengujian, dilakukan perbaikan terhadap *Game Math & Trash*, pada penambahan peringatan reset level dimenu *Leaderboard*.

#### 3. Hasil dan Pembahasan

##### 3.1. Hasil

Aplikasi permainan "Math & Trash" ini telah dikembangkan sesuai dengan System Development Life Cycle. Tahapan yang berhasil dilalui mulai dari analisis kebutuhan, spesifikasi, desain, implementasi sampai pengujian. Implementasi perangkat lunak menggunakan aplikasi Construct 2 yang merupakan game builder berbasis HTML 5. Tahapan pengujian

meliputi uji verivikasi white box testing dan black box testing. Kemudian tahapan uji validasi portability.

Aplikasi permainan "Math & Trash" yang telah dikembangkan memiliki kelebihan dan kekurangan. Adapun kelebihan dan kekurangan aplikasi ini antara lain:

Kelebihan aplikasi permainan "Math & Trash"

1. Aplikasi ini memberikan materi tentang perbedaan sampah organik dan non organik.
2. Aplikasi ini dapat berjalan pada berbeda-beda versi android.
3. Aplikasi ini dapat berjalan pada beberapa *platform* android dengan ukuran layar yang berbeda.
4. Aplikasi ini mampu menyimpan data skor yang telah berhasil pemain dapatkan.

Kekurangan apliksi permainan "Math & Trash":

1. Aplikasi ini memiliki pertanyaan yang tidak acak.
2. Aplikasi ini tidak *support* rotasi layar.

##### 3.2. Pembahasan

Tabel 6. Hasil penilaian anak-anak pada saat uji coba

No	Jenis Pertanyaan	Rerata Skala	Kriteria
1	Tampilan desain <i>game</i>	4,7	Sangat Baik
2	Kesamaan bentuk tombol	4,8	Sangat Baik
3	Ketepatan pemilihan music	4,8	Sangat Baik
4	Ketepatan pemilihan suara dan gambar	4,8	Sangat Baik
5	Ketepatan ukuran gambar	4,7	Sangat Baik
6	Kemudahan penggunaan <i>game</i>	4,8	Sangat Baik
7	Kejelasan bantuan pada <i>game</i>	4,8	Sangat Baik
8	Tampilan teks jelas dan tidak berlebihan	4,8	Sangat Baik
9	Kemudahan penggunaan tombol	4,7	Sangat Baik
10	Pengaturan ON/OFF music	4,8	Sangat Baik
11	Kesesuaian dengan kegunaan	4,8	Sangat Baik
12	Siswa dan <i>game</i> dapat berinteraksi	4,8	Sangat Baik
13	Siswa dapat melakukan operasi penambahan, pengurangan,	4,7	Sangat Baik

	perkalian, dan pembagian		
14	Siswa memahami konsep dasar matematika	4,7	Sangat Baik
15	Siswa dapat mengerti perbedaan sampah organik dan sampah non organik	4,7	Sangat Baik
16	Kesulitan <i>game</i> sesuai dengan kemampuan siswa	4,8	Sangat Baik
	Total Keseluruhan	<b>76,2</b>	<b>Sangat baik</b>
	Rata-rata	4,7625	

Dari table 6 dapat disimpulkan bahwa *game* edukasi yang dibuat termasuk dalam kriteria “Sangat baik”. Dengan rata-rata skor sebesar 4.7.

#### Hasil Akhir Produk

Aplikasi permainan “*Math & Trash*” ini telah dikembangkan sesuai dengan *Game Development Life Cycle*. Tahapan yang berhasil dilalui mulai dari analisis kebutuhan, spesifikasi, desain, implementasi sampai pengujian. Implementasi perangkat lunak menggunakan aplikasi Construct 2 yang merupakan *game builder* berbasis HTML 5. Tahapan pengujian meliputi uji verifikasi *white box testing* dan *black box testing*. Kemudian tahapan uji validasi *portability*.

Aplikasi permainan “*Math & Trash*” yang telah dikembangkan memiliki kelebihan dan kekurangan. Kelebihannya adalah aplikasi ini memberikan materi tentang perbedaan sampah organik dan non organik. Dapat berjalan pada versi android berbeda, dan *platform* android dengan ukuran layar yang berbeda. Mampu menyimpan data skor yang telah berhasil pemain dapatkan.

Kekurangannya, aplikasi ini memiliki pertanyaan yang tidak acak dan tidak *support* rotasi layar.

#### 4. Kesimpulan

Perancangan *Game* “*Math & Trash*” menggunakan metode GDLC, yaitu *preproduction*, *production*, *testing* dan *postproduction*. Hasil aspek *portability Game* “*Math & Trash*” kategori Sangat Baik, aplikasi mampu berjalan diberbagai versi android dan berbagai ukuran layar yang berbeda. Berdasarkan hasil rata-rata pengujian aplikasi “*Math & Trash*” dengan uji *whitebox* dan *blackbox* didapatkan hasil persentase sebesar 100%, maka dari hasil tersebut *Game* “*Math & Trash*” memenuhi kategori kelayakan Sangat Layak. Berdasarkan hasil dari uji kelompok kecil dengan menggunakan kuisioner dihasilkan data nilai dengan rata-rata skor (x) sebesar 4.7 dengan termasuk kriteria Sangat Baik. Berdasarkan hasil dari responden sesudah dan sebelum menggunakan *game* tidak ada nilai negative (-) yang artinya ada perubahan setelah memainkan *game* edukasi tersebut dan berpengaruh Baik.

#### Daftar Rujukan

[1] M. R. Rahadi, K. I. Satoto, and I. P. Windasari, “Perancangan Game Math Adventure Sebagai Media Pembelajaran Matematika Berbasis Android,” *J. Teknol. dan Sist. Komput.*,

vol. 4, no. 1, p. 44, 2016.  
 [2] A. Vega, “Game edukasi sebagai media pembelajaran pendidikan anak usia dini,” *Inform.*, vol. 1, no. 1, pp. 1–8, 2016.  
 [3] P.-M. Noemi and S. H. Máximo, “Educational games for learning,” *Univers. J. Educ. Res.*, vol. 2, no. 3, pp. 230–238, 2014.  
 [4] A. Budianto, “Analisis Dan Perancangan Game Edukasi ‘Need for Safety’ Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas Untuk Anak Usia 6-12 Tahun,” *J. Pendidik. Tek. Inform.*, pp. 1–7, 2014.  
 [5] A. F. Chow, K. C. Woodford, and J. Maes, “Deal or no deal: Using games to improve student learning, retention and decision-making,” *Int. J. Math. Educ. Sci. Technol.*, vol. 42, no. 2, pp. 259–264, 2011.  
 [6] T. official reference from Scirra, *User Manual Construct 2*. scirra.com, 2019.  
 [7] W. C. S., “Rancang Bangun Game Edukasi Matematika Kelas 4 - 6 Sd Berbasis Android Menggunakan Construct 2,” UN PGRI Kediri :, 2017.  
 [8] D. N. Molina, Melia, and H. Sopryadi, “Rancang Bangun Edugame Untuk Pembelajaran Profil Negara-Negara Asean Berbasis Android,” STIMIK GI MDP. Palembang., 2014.  
 [9] G. Kiryakova, N. Angelova, and L. Yordanova, “Gamification in Education,” *Proc. 9th Int. Balk. Educ. Sci. Conf.*, pp. 1–5, 2014.  
 [10] C. Aliferi, *Android Programming Cookbook*. 2016.  
 [11] W. A. Rifai, “Pengembangan Game Edukasi Lingkungan Berbasis Android,” Universitas Negeri Yogyakarta, 2015.  
 [12] G. Play, “Top categories Most popular Google Play categories,” 2019. [Online]. Available: <https://www.appbrain.com/stats/android-market-app-categories>.  
 [13] R. C. Putra, “Pembuatan Game Edukasi Pintar Memilih Sampah Berbasis Android,” Universitas Muhammadiyah Surakarta, 2016.  
 [14] B.-S. Indonesia, *Statistik Lingkungan Hidup Indonesia Environment Statistics of Indonesia 2017*. Badan Pusat Statistik/BPS – Statistics Indonesia, 2017.  
 [15] M. Ahsan and I. Faud, “Game Edukasi Memilih Sampah Berbasis Android Menggunakan Algoritma a-Star (a\*),” *JIMP - J. Inform. Merdeka Pasuruan*, vol. 3, no. 1, pp. 1–13, 2016.  
 [16] M. Adiwijaya, K. I. S., and Y. Christyono, “Perancangan Game Edukasi Platform Belajar Matematika Berbasis Android Menggunakan Construct 2,” *Transient*, vol. 4, no. 1, pp. 128–133, 2015.  
 [17] I. D. Putu, A. Sudiatmika, A. A. K. A. Cahyawan, and P. W. Buana, “Aplikasi Game Edukasi Trash Grabber Untuk Mengenal Jenis-Jenis Sampah Pada Smartphone Berbasis Android,” *Merpati*, vol. 2, no. 2, pp. 215–225, 2014.  
 [18] Pemerintah Republik Indonesia, “Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional,” 2003.  
 [19] I. G. Rasagama, “Memahami Implementasi ‘Educational Research and Development,’” Bandung, 2011.  
 [20] H. Maxwell Candler, *The Game Production Handbok third ed.*, 3rd ed. Burlington, MA 01803: Jones and Bartlett Publishers, 2013.  
 [21] P. Nguyen, “Game Production and Role of Game Producer,” SAVONIA UNIVERSITY OF APPLIED SCIENCES, 2014.  
 [22] R. Ramadan and Y. Widyani, “Game development life cycle guidelines,” in *2013 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2013*, 2013, no. September 2013, pp. 95–100.  
 [23] L. Husniah, B. F. Pratama, and H. Wibowo, “Gamification

- And GDLC (Game Development Life Cycle) Application For Designing The Sumbawa Folklore Game "The Legend Of Tanjung Menangis (Crying Cape)," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 3, no. 4, p. 351, 2018.
- [24] Romeo, *Testing dan implementasi sistem*, 1st ed. Surabaya: STIKOM Surabaya, 2003.
- [25] H.-E. Eriksson, M. Penker, B. Lyons, and D. Fado, *UML™ 2 Toolkit*. Indianapolis, Indiana: Wiley Publishing, Inc., 2004.
- [26] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*. Canada.: Addison Wesley Longman, Inc., 1999.
- [27] R. S. Pressman, *Rekayasa Perangkat Lunak (Pendekatan Praktisi)*, 7th ed. AndiPublisher, 2012.
- [28] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*, 22nd ed. Bandung: Alfabeta Bandung, 2015.
- [29] F. G. H. Saputro, "Rancang Bangun Game Pertualangan Satwa Langka Di Indonesia Menggunakan Construct 2 Berbasis Android," Universitas Stikubank (UNISBANK) Semarang, 2017.
- [30] R. S. Pressman, *Software Engineering A Practitioner's Approach 5th ed.* 2001.
- [31] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "Penggunaan Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)," *J. Ilm. Teknol. Inf. Terap.*, vol. I, no. 3, pp. 31–36, 2015.
- [32] S. S. Ghuman, "Software Testing Techniques," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 10, pp. 988–993, 2014.