Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Hasil penelitian terdahulu dijadikan sebagai referensi dalam melakukan penelitian ini, dalam penelitian tersebut terdapat kesamaan yang dijadikan acuan untuk melakukan perancangan aplikasi ini.

Tabel 2.1 Penelitian Terdahulu

No.	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1.	Surya, KP., 2017.	"Perancangan dan Pembuatan *Game* Racoon Run Dengan Menggunakan Construct 2"	Pembuatan *game* dengan konsep *side scrolling endless run* dapat dibuat dengan menggunakan karakter yang berlari ke satu arah untuk mengumpulkan skor sebanyak-banyaknya.
	Perbedaan: penelitian yang dilakukan oleh Surya pada tahun 2017 membahas terkait *game* yang dirancang melalui construct 2 dengan konsep *side-scrolling endless run*		
2.	T. Satria, A. F. Rochim, and I.P. Windasari, 2015.	"Perancangan Aplikasi Pembelajaran 'Fruvenimal' Berbasis HTML5"	Perancangan *game* ini meningkatkan pengetahuan keanekaragaman buah, sayur, dan hewan pada anak
	Perbedaan: penelitian yang dilakukan oleh T. Satria, A. F. Rochim, and I. P. Windasari, pada tahun 2015 membahas terkait *game* yang dirancang melalui construct 2 dengan tujuan mencakup pengetahuan terkait pengenalan terhadap sejumlah buah, sayur, dan hewan tertentu.		

Rancang Bangun Permainan..., Muhammad Yuki Ramadhan, FTI UMN, 2019

Meminimalisasikan koding untuk membuat *game*, lalu ada poin di mana pembahasan terkait membuat *game* dengan menggunakan *Construtct 2* untuk menjalankan di *multi-plaftorm* dengan mengambil kesimpulan bahwa dengan menggunakan *Construct 2* seseorang yang memiliki pengetahuan pemrograman minim dapat membuat *game* sesuai kehendaknya. (Surya, 2017)

Dijelaskan bahwa *game* efektif untuk meningkatkan pengetahuan keanekaragaman buah, sayur, dan hewan pada anak. Diberikan saran dalam upaya pengembangan aplikasi berupa memperluas cakupan aplikasi meliputi buah, sayur, dan hewan lainnnya. (Satria, Rochim, & Windasari, 2015)

Dari kedua penelitian terinisiasi untuk melakukan perancangan *game* bersifat edukatif yang mengajarkan anak terkait pengetahuan balita 0-6 tahun untuk memperkenalkan angka dan huruf, kedua peneliti menggunakan *tools* yang sama yaitu Construct 2 dengan perbedaan memberikan *platform* sebagai media pembelajaran anak melalui *handphone*, yang diharapkan akan membantu orang tua untuk memberi edukasi terkait pengenalan angka dan huruf yang salah satunya mendasari keseluruhan bidang pendidikan anak di tingkat selanjutnya.

2.2. Game

Untuk memahami *game*, Anda harus memiliki ide tentang apa itu *game*. Ini adalah bisnis definisi: untuk membuat batasan ide dan fenomena sehingga Anda dapat lebih tahu apa yang sedang dibahas. (Arjoranta, 2014).

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.3. Game Edukasi

Game edukasi adalah salah satu jenis media yang digunakan untuk memberikan pengajaran, menambah pengetahuan penggunanya melalui suatu media unik dan menarik. *Game* edukasi dibuat dengan tujuan spesifik sebagai alat pendidikan, untuk belajar mengenal warna, mengenal huruf dan angka, matematika, sampai belajar bahasa asing (Rifai, 2015).

Game edukasi merupakan permainan yang dikemas untuk merangsang daya pikir dan termasuk salah satu cara untuk melatih meningkatkan konsentrasi penggunanya (anak-anak). Pemanfaatan teknologi *Game* edukasi pada proses belajar mengajar anak merupakan salah satu cara yang tepat, karena *Game* edukasi sebagai media visual memiliki kelebihan dibandingkan dengan media visual yang lain. (Rahman & Tresnawati, 2016).

Sebuah media pembelajaran yang bersifat mendidik, dimana dengan media tersebut dapat mendorong siswa untuk berpikir kreatif dan melakukan kegiatan dengan sesama siswa dalam melakukan permainan dalam kegiatan pembelajaran (Suwarno, 2017). Jenis game edukasi sebenarnya lebih menekankan pada rancangan fungsional, bukan jenis pada rancangan teknis. (Pratama & Haryanto, 2017).

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.4. Pendidikan Anak Usia Dini

Pendidikan anak usia dini (PAUD) adalah jenjang pendidikan sebelum jenjang pendidikan dasar yang merupakan suatu upaya pembinaan yang ditujukan bagin anak sejak lahir sampai dengan usia enam tahun yang dilakukan melalui pemberian rangsangan pendidikan untuk membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lebih lanjut, yang diselenggarakan pada jalur formal, nonformal, dan informal. (Hermoyo, 2015)

Anak usia dini merupakan saat seseorang mengalami perkembangan dan pertumbuhan yang pesat dalam hidupnya. Oleh karena itulah anak usia dini membutuhkan beragam stimulus yang dapat membantunya untuk berkembang dengan baik sesuai dengan kebutuhan dan potensinya. (Alia & I., 2018).

2.5. Construct 2

Dikutip dari website resmi www.scirra.com, *Construct* 2 adalah sebuah 2D *game editor* berbasis HTML 5, dirancang oleh Scirra.Ltd. Perangkat lunak ini mengincar masyarakat yang ingin membuat *game* tetapi tanpa *basis* seorang *programmer* atau memiliki pengetahuan *programming* yang minim.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 2.1 Logo Construct 2

Construct 2 memungkinkan seseorang membuat *game* hanya dengan *drag and drop* dan menggunakan *visual editor* dan *behavior-based logic system.*

Construct 2 dibuat oleh Scirra. Ltd yang telah berdiri dari tahun 2011 oleh Ashley dan Thomas Gullen bersaudara. Program pertama mereka adalah *construct classic*, yang dikembangkan sejak tahun 2007. *Construct classic* membuat permainan berbasis DirectX 9 untuk *platform windows* dan sekarang menjadi construct 2 yang dapat membuat *game* dengan banyak *platform* contohnya adalah HTML 5, iOS, Android, Linux, Windows (Scirra, 2019).

2.6. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak (Putra, 2012).

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Android adalah sebuah sistem operasi untuk perangkat lunak mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Sistem operasi Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi. (Widiyanto & Nuryanto, 2015).

2.7. Metode *Iterative Game Development*

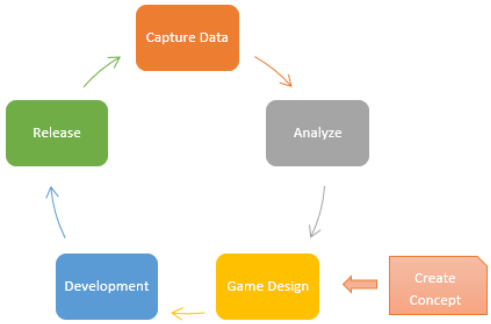
Metode *Iterative* ini akan memproses *game* secara terus menerus karena sebuah *game* akan selalu mengalami perubahan di setiap bagiannya. *Game* akan selalu memiliki usulan, purwarupa, *running test*, dan akan selalu ada evaluasi saat *game* sudah rilis ataupun belum rilis (Surya, 2017).

Metode *iterative* memiliki prinsip kerja yaitu tidak realistik jika membuat sebuah *game* yang ideal dan diinginkan hanya dalam satu kali percobaan. Dengan membuat dan mengetes inti dari kriteria *game* (misalnya kesenangan pengguna saat bermain), desainer *game* akan mampu meningkatkan potensi pasaran *game* yang telah dibangun (Surya, 2017).

Metode *iterative* dapat diterapkan dibidang mana saja, tetapi metode ini sangat berpengaruh besar dalam pembuatan *game*.

Game desainer membuat purwarupa dan menjalan kan testing untuk mengukur game element functionality dan memperluas dan menemukan konsep game yang baru untuk meningkatkan kepuasan pengguna.

Rancang Bangun Permainan..., Muhammad Yuki Ramadhan, FTI UMN, 2019



Gambar 2.2. Metode Iterative Game Development

Berikut merupakan penjelasan dari masing-masing tahapan pada metode iterative game development:

a. Membuat Konsep Game

Proses untuk membuat dan memikirkan konsep *game* yang akan dibuat, contohnya *game* FPS (*First Person Shooter*), MOBA (*Multiplayer Online Battle Arena*), *board game*, atau jenis lainnya.

b. Desain *Game*

Proses ini adalah untuk membuat desain dari *game* yang sudah ditentukan konsepnya. Pada tahap ini, *developer* membuat purwarupa dari *game* yang ingin dikembangkan lebih lanjut.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

c. Development

Pada proses ini *developer* akan membuat *game* yang sudah memiliki konsep dan purwarupa yang pada proses sebelumnya dilakukan, *developer* akan membuat *game* sampai *game* tersebut selesai dan siap diluncurkan ke pasaran.

d. Release

Tahap ini adalah tahap di mana *developer* siap meluncurkan *game* yang telah selesai dibuat ke pasaran agar dapat digunakan oleh pengguna.

e. Capture Data

Pada tahap ini *developer* akan mencari umpan balik atau komentar dari para pengguna *game* yang sudah dirilis.

f. Analisis *Data Game*

D*eveloper* akan menggunakan umpan balik, komentar, kritik dan saran dari para pengguna agar *developer* dapat mencari, memperbaiki, dan mengembangkan *game* agar dapat diterima oleh pengguna.

2.8. *Black Box Testing*

Black Box Testing adalah bentuk pengujian fungsionalitas di mana fungsi kritis dipertimbangkan. Oleh karena itu, jenis tes ini tidak memerlukan pengetahuan khusus tentang implementasi yang mendasarinya (Suthar, 2015).

Dalam black box testing, seorang tester tanpa pengetahuan tentang struktur internal suatu aplikasi dapat menguji aplikasi tersebut, memberikan input, dan memverifikasi output terhadap hasil yang diharapkan. (Suthar, 2015).

Rancang Bangun Permainan..., Muhammad Yuki Ramadhan, FTI UMN, 2019

Black box testing juga dikenal sebagai *behavioral testing*, adalah metode pengujian perangkat lunak di mana struktur internal / desain / implementasi dari item yang diuji tidak diketahui oleh tester. Jenis defect yang ditemukan oleh Black Box Testing (Chavhan, 2015):

Defect yang dapat ditemukan oleh *black box testing* yang dijelaskan oleh (Chavhan, 2015) sebagai berikut :

- Mengidentifikasi fungsi yang hilang dalam sistem.
- Mengobservasi *Usability Issues* selama proses *black box testing.*
- Menemukan *Performed Issues* ketika melakukan tes sesuai kapasitas.
- Membantu mengidentifikasi *application behaviour* ketika banyak pengguna menggunakan pada saat yang bersamaan.

2.9. Use Case Diagram

Use case dideskripsikan secara tekstual dalam bentuk use case scenario untuk menjelaskan interaksi yang terjadi antara aktor dengan sistem. Selanjutnya, use case diilustrasikan secara visual dalam bentuk use case diagram untuk menggambarkan konteks dari sistem yang dikembangkan. Dalam praktiknya, kedua model tersebut tidak sulit untuk dibuat meskipun oleh orang yang belum berpengalaman. Namun demikian, pemodelan use case yang dihasilkan, baik dalam konteks pembelajaran konsep pengembangan perangkat lunak di kampus maupun dalam konteks implementasi di industri perangkat lunak, tidak sedikit yang mengandung kesalahan baik secara sintaksis maupun semantik (Kurniawan, 2018).

Rancang Bangun Permainan..., Muhammad Yuki Ramadhan, FTI UMN, 2019

Salah satu diagram penting yang digunakan untuk mengilustrasikan kebutuhan (*requirements*) dari sistem adalah use case (UC) diagram, yang menjelaskan secara visual konteks dari interaksi antara aktor dengan sistem. Setiap use case menyatakan spesifikasi perilaku (fungsionalitas) dari sistem yang sedang dijelaskan yang memang dibutuhkan oleh aktor untuk memenuhi tujuannya (Kurniawan, 2018).

Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan UC. Aktor adalah segala sesuatu yang berinteraksi langsung dengan sistem, bisa merupakan orang (yang ditunjukkan dengan perannya dan bukan namanya/personilnya) atau sistem komputer yang lain. Aktor dinotasikan dengan simbol gambar orang-orangan (*stick-man*) dengan nama kata benda di bagian bawah yang menyatakan peran/sistem. Aktor bisa bersifat primer, yaitu yang menginisiasi berjalannya sebuah UC, atau sekunder, yaitu yang membantu berjalannya sebuah UC. UC dinotasikan dengan simbol elips dengan nama kata kerja aktif di bagian dalam yang menyatakan aktivitas dari perspektif aktor. Setiap aktor dimungkinkan untuk berinteraksi dengan sistem dalam banyak UC. Sebaliknya, setiap UC bisa dijalankan oleh lebih dari satu aktor. (Kurniawan, 2018).

Hanya bagian yang ditarik pada *use case diagram* adalah batas sistem, use case itu sendiri, aktor, dan berbagai asosiasi antara komponen-komponen ini. Kekuatan utama *use case diagram* adalah menyediakan gambaran umum kepada pengguna tentang proses bisnis. Ada empat jurusan langkah-langkah dalam menggambar *use case diagram* (Dennis , Wixom , & Tegarden , 2015)

Rancang Bangun Permainan..., Muhammad Yuki Ramadhan, FTI UMN, 2019

Pertama, kita menempatkan dan menggambar use case pada diagram. Ini diambil langsung dari jurusan gunakan kasing yang sebelumnya diidentifikasi. Asosiasi spesial case-use *(include, extend,* atau *generalization*) juga ditambahkan ke model pada titik ini. Hati-hati dalam menyusun diagram. Ada tidak ada pesanan formal untuk kasus penggunaan, sehingga mereka dapat ditempatkan dengan cara apa pun yang diperlukan untuk membuat diagramnya mudah dibaca dan meminimalkan jumlah garis yang bersilangan. Itu sering diperlukan untuk menggambar ulang diagram beberapa kali dengan menggunakan kasus di tempat yang berbeda untuk membuat diagram mudah dibaca. Juga, untuk tujuan dimengerti, seharusnya tidak lebih dari tiga hingga sembilan use case pada model menghitung use case yang telah diperhitungkan dan sekarang terkait dengan use case lain melalui hubungan include, extended, atau generalisasi. Kedua, para aktor ditempatkan dan digambarkan pada diagram. Untuk meminimalkan jumlah garis bahwa lintas pada diagram, para aktor harus ditempatkan dekat kasus penggunaan dengan siapa mereka berada terkait. Ketiga, batas subjek ditarik. Ini membentuk batas subjek, memisahkan penggunaan kasing (mis., fungsi subjek) dari aktor (mis., peran pengguna eksternal). Langkah keempat dan terakhir adalah menambahkan asosiasi dengan menggambar garis untuk menghubungkan para aktor dengan penggunaan kasus-kasus dimana mereka berinteraksi. Tidak ada urutan yang ditunjukkan oleh diagram, dan item ditambahkan di sepanjang cara tidak harus ditempatkan dalam urutan tertentu; karena itu, mungkin membantu mengatur ulang simbol sedikit untuk meminimalkan jumlah garis yang bersilangan, membuat diagram kurang membingungkan. (Dennis , Wixom , & Tegarden , 2015).

Rancang Bangun Permainan..., Muhammad Yuki Ramadhan, FTI UMN, 2019

2.10. Kerangka Berpikir

Kerangka Berpikir merupakan model konseptual tentang teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting. Kerangka berpikir yang baik akan menjelaskan secara teoritis pertautan antar variabel yang akan diteliti. jadi secara teoritis perlu dijelaskan hubungan antar variabel indenpenden dan dependen (Sugiyono, 2017).

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA